ECE 564- Spring 2007

Control and Operation of Electric Power Systems

Midterm Exam (Take Home)

March 22, 2007

Kathleen E Williams

Kathleen E. Williams

Problem # 1

Assume the fuel consumption functions of the three units are given by:

$$F_1(P_1) = 10 + 6P_1 + 0.0050P_1^2 \; MBtu/h$$
$$F_2(P_2) = 20 + 8P_2 + 0.0025P_2^2 \; MBtu/h$$
$$F_3(P_3) = 30 + 9P_3 + 0.0020P_3^2 \; MBtu/h$$

Solve the unit commitment problem by Lagrange Relation method and iterate until two distinct feasible primal solutions are obtained. Show duality gap in each iteration. Assume $\alpha_\lambda^+ = 0.05$, $\alpha_\mu^+ = 0.05$, $\alpha_\lambda^- = 0.01$, $\alpha_\mu^- = 0.01$.

Problem Formulation

See tables 1 and 2 for unit parameters and system load and reserve requirements

Assumptions: Problem does not minimize CO_2 emissions, only costs. Problem DOES consider minimum up/down time constraints, unit power limits, and reserve requirements.

t = *hour*
F_i = *cost*
S_i = *startup cost*
i = *unit*

Dual Function

$$p(t,i) = \frac{\lambda^t - b(i)}{2c(i)}$$

$$\mathcal{L} = \sum_{t=1}^{4}\sum_{u=1}^{3}\left[F_i u_i^t + S_i u_i^t\left(1 - u_i^{t-1}\right)\right] + \sum_{t=1}^{4}\lambda^t\left(P_{LOAD}{}^t - \sum_{i=1}^{3} P_i^t u_i^t\right) + \sum_{t=1}^{4}\mu^t\left(P_{LOAD}{}^t + R^t - \sum_{u=1}^{3} P_i^{\max} u_i^t\right)$$

$$q = \mathcal{L}$$
$$q^* = \max \mathcal{L}$$
$$subject\ to\ p_i^{\min} \le p_i \le p_i^{\max}$$

$$X(t+1, i) = \max\{X(t, i) - 1, T^{OFF}\} \quad T^{OFF} = minimum\ downtime\ (hours)$$
$$X(t+1, i) = \min\{X(t, i) + 1, T^{ON}\} \quad T^{ON} = minimum\ uptime\ (hours)$$

Let initial values for λ and μ be equal to zero for all time periods.

We calculate P_i and u_i in each sub problem. Correct for minimum up/down times constraints.

$$\boxed{P_i^{\max} u_i \geq P_{LOAD}{}^{t} + R^t}$$

u_1, P_1 $\quad$ u_2, P_2 $\quad$ u_3, P_3

λ^t, μ^t $\quad$ λ^t, μ^t $\quad$ λ^t, μ^t

$$\boxed{\min \sum_{t=1}^{4} \begin{bmatrix} F_1 u_1^t + S_1 u_1^t \left(1 - u_1^{t-1}\right) \\ -\lambda^t P_1^t u_1^t - \mu^t P_1^{\max} u_1^t \end{bmatrix}} \quad \boxed{\min \sum_{t=1}^{4} \begin{bmatrix} F_2 u_2^t + S_2 u_2^t \left(1 - u_2^{t-1}\right) \\ -\lambda^t P_2^t u_2^t - \mu^t P_2^{\max} u_2^t \end{bmatrix}} \quad \boxed{\min \sum_{t=1}^{4} \begin{bmatrix} F_3 u_3^t + S_3 u_3^t \left(1 - u_3^{t-1}\right) \\ -\lambda^t P_3^t u_3^t - \mu^t P_3^{\max} u_3^t \end{bmatrix}}$$

For each hour t, only one λ_2^t, μ^t is used by each subproblem.

$X_i^t > 0$ cumulative number of hours that unit i has been on.

$X_i^t < 0$ cumulative number hours that unit i has been off.

Calculate dual solution

$$q^* = \sum_{t=1}^{4} \sum_{u=1}^{3} \left[F_i u_i^t + S_i u_i^t \left(1 - u_i^{t-1}\right) \right] + \sum_{t=1}^{4} \lambda^t \left(P_{LOAD}{}^{t} - \sum_{i=1}^{3} P_i^t u_i^t \right) + \sum_{t=1}^{4} \mu^t \left(P_{LOAD}{}^{t} + R^t - \sum_{u=1}^{3} P_i^{\max} u_i^t \right)$$

where P_i are dual values.

Does capacity satisfy system load and reserve?

If Yes → Primal Problem
If No→ Update λ and μ

Continue to next iteration.

Primal Problem

If $P_i^{\max} u_i^t \geq P_{LOAD}^{(t)} + R^t$ for hour t

Given u_i, unit commitments for hour t

Do economic dispatch by λ - iteration :

`lamedc=[lamsmin(t)+lamsmax(t)]/2`

Use `Pi =[lamedc-b(i)/2c(i)]` where P_i is the primal P_i

$$J^* = \sum_{t=1}^{4}\sum_{u=1}^{3}\left[F_i u_i^t + S_i u_i^t\left(1-u_i^{t-1}\right)\right] \text{ where } S_i = \text{startup costs}$$

Duality Gap

$$\text{Duality Gap} = \frac{J^* - q^*}{q^*}$$

If dual < 0, then gap = 50,000

If duality gap $< \varepsilon$, consider solution.

I chose ε to be 0.1 for this problem.

Other Formulations

$$\sum_{t=1}^{4}\frac{\partial q}{\partial \lambda} = \sum_{t=1}^{4} P_{LOAD}^{(t)} - \sum_{t=1}^{4}\sum_{u=1}^{3} P_i$$

where P_i is the dual P_i

$$\lambda^{t+1} = \lambda^t + \alpha\frac{\partial q}{\partial \lambda^t}$$

$$\frac{\partial q}{\partial \lambda^t} = P_{LOAD}^{(t)} - \sum_{u=1}^{3} P_i^t$$

$$if\ \frac{\partial q}{\partial \lambda} > 0,\ then\ \alpha = 0.05$$

$$if\ \frac{\partial q}{\partial \lambda} < 0,\ then\ \alpha = 0.01$$

$$\sum_{t=1}^{4}\frac{\partial q}{\partial \mu} = \sum_{t=1}^{4} P_{LOAD}^{(t)} + \sum_{t=1}^{4} Reserve^{(t)} - \sum_{t=1}^{4}\sum_{u=1}^{3} P_i$$

where P_i is the dual P_i

$$\mu^{t+1} = \mu^t + \alpha\frac{\partial q}{\partial \mu^t}$$

$$\frac{\partial q}{\partial \mu^t} = P_{LOAD}^{(t)} + Reserve^{(t)} - \sum_{u=1}^{3} P_i^t$$

$$if\ \frac{\partial q}{\partial \mu} > 0,\ then\ \alpha = 0.05$$

$$if\ \frac{\partial q}{\partial \mu} < 0,\ then\ \alpha = 0.01$$

Results

Iteration	Duality Gap
1	50000.0
2	3.1575
3	1.4056
4	0.70179
5	0.34615
6	0.097004
7	0.2455
8	182.16
9	1.2661
10	0.0076838
11	0.48696
12	0.26174
13	0.11848
14	0.40616
15	108.67
16	62.15
17	0.46821
18	0.44602
19	114.46
20	0.47806
21	0.076618
22	480.34
23	50000.0
24	3.6241
25	0.41089
26	1.0696
27	0.52289
28	0.34321
29	39.204
30	173.54
31	0.58286
32	0.25895
33	35.627
34	0.083243
35	0.62972
36	0.19819
37	64.462
38	0.63674
39	0.97103
40	0.73307
41	0.73041
42	0.70174
43	97.189
44	31.784
45	0.17761
46	0.78349
47	99.378
48	0.72543
49	0.3475
50	46.97

Iteration	Duality Gap
51	0.59239
52	0.801
53	1.1981
54	0.75252
55	0.38536
56	0.77468
57	91.434
58	0.8308
59	186.99
60	0.77746
61	84.931
62	0.78262
63	0.43707
64	0.85183
65	96.82
66	22.345
67	0.82791
68	0.80421
69	0.017026
70	76.558
71	1.3383
72	0.81138
73	166.56
74	19.715
75	0.6859
76	16.274
77	0.063075
78	0.87555
79	0.57506
80	13.764
81	18.015
82	1.3101
83	0.84911
84	98.834
85	0.83834
86	126.69
87	0.84445
88	348.59
89	0.83935
90	2.086
91	0.84901
92	0.69194
93	0.85347
94	170.57
95	13.03
96	0.91539
97	0.87018
98	66.992
99	0.85818
100	1.1703

Two Distinct Feasible Solutions

Solution # 1

dualitygap =

0.0398

dualP =

0	300.0	125.0
317.2	234.4	50
389.44	378.88	200.00
280.0	160.0	0

primalP =

0	244.44	55.55
316.66	233.32	50
354.54	309.08	136.35
266.66	133.32	0

units =

0	1	1
1	1	1
1	1	1
1	1	0

dualf =

85924

primalf =

89340

Solution #2

dualitygap =

0.017026

dualP =

0	400	200
600	400	200
600	400	200
600	400	200

primalP =

0	244.44	55.55
316.66	233.32	50
354.54	309.08	136.35
249.98	100	50

units =

0	1	1
1	1	1
1	1	1
1	1	1

dualf =

91172

primalf =

89619

```
---------------------------------------------------------------------------
 ECE 564    Spring 2007
 Exam # 1 -Midterm
 Due March 22, 2007
 Kathleen E. Williams

 Problem #1 Unit Commitment with Quadratic Fuel Consumption Function
---------------------------------------------------------------------------

echo off
clear

 fuel consumption parameters times the price ($5.0/MBTU)
a=[10,20,30]*5;
b=[6,8,9]*5;
c=[0.005,0.0025,0.002]*5;

 hourly load and reserve
pload=[300,600,800,400];
reserve=[15,30,40,20];

 units min, max values
pmin=[100,100,50];
pmax=[600,400,200];

 startup costs of units
startup=[1000,500,0];

 min down-time, up-time, and initial conditions of units
minon=[3,2,1];
minoff=[-3,-2,-1];
init=[-2,-3,-2];

 determines lambda search range for economic dispatch in primal problem
for j=1:3
    lammin(j) = b(j)+ 2*c(j)*pmin(j);
    lammax(j) = b(j)+ 2*c(j)*pmax(j);
end

 initializes the lamda and mu dual variables for the Lagrange relaxation
lamdual=[0,0,0,0];
mudual=[0,0,0,0];

 keeps track the duality gap in each iteration
dualgap = [];

---------------------------------------------------------------------------
 Lagrange Relaxation Loop

 Comments: Instead of stopping the loop by the value of the duality gap
 I chose to iterate multiple times to find multiple gaps and hence,
 multiple unit commitment solutions
---------------------------------------------------------------------------

for i=1:100

 initialize X

 Comments: X holds the unit up/downtime. For example, X(t,j) means the
 min up/down time at hour t and unit j. If X(t,j) were equal to 2, that
 means the unit has been online for 2 hours. If X(t,j) were equal to -2,
 that means the unit has been offline for two hours
 t = hour
 j = unit

for t=1:4
   for j=1:3
       X(t,j) = 0;
       if t == 1
           X(t,j) = init(j);   takes initial conditions for hour 1
```

```
        else
        end
    end
end


      ------------------------------------------------------------------------
      Dual Loop

      This begins the dual function calculation

    for t=1:4

        for j=1:3

             Determine Pi at hour t given dual lambda and unit i parameters
           p(t,j)=(lamdual(t)-b(j))/(2*c(j));

             Check Pi dual solution at min and max limits
           if p(t,j) < pmin(j)
               p(t,j) = pmin(j);
           end
           if p(t,j) > pmax(j)
               p(t,j) = pmax(j);
           end


              Special Condition is at Hour 1
            if t==1

                  Unit Commitment

                  Comments: Determine commitment of unit i at hour t given
                  dual values of lambda and mu at minimal

                if ((a(j) + b(j)*p(t,j) + c(j)*p(t,j)^2)) + startup(j) -
lamdual(t)*p(t,j)-mudual(t)*pmax(j) < 0
                    u(t,j) = 1;
                else
                    u(t,j) = 0;
                end;


                  Minimum Up/Down Time Constraints

                  Comments: Use the X(t,j) variable to compare against
                  unit minimum up/down times. If constraint is not
                  satisfied, the unit commitment must be the same as the
                  previous hour until minimum up/down time is satisfied

                  Since this is the first hour, we know the units are
                  initially offline, so we need only consider two
                  possibilities: turning the unit on, or off
                if u(t,j) == 1
                    if X(t,j) > minoff(j)
                          Since the minimum off time is violated, keep
                          the unit offline!
                        u(t,j)= 0;
                          Increment the unit offtime and update X(t,j)
                        X(t+1,j) = max(X(t,j)-1,minoff(j));
                    else
                          The unit does NOT violate minimum off time, keep
                          the unit committed and increment the unit ontime
                          and update X(t,j)
                        X(t+1,j) = min(X(t,j)+1,minon(j));
                    end

                else
                      The unit i is not committed, so we only need to
```

```
            Increment the unit offtime and update X(t,j)
        X(t+1,j) = max(X(t,j)-1,minoff(j));
    end



    Considering all other hours t > 1

else

        Unit Commitment

        Comments: Determine commitment of unit j at hour t given
        dual values of lambda and mu at minimal

    if ((a(j) + b(j)*p(t,j) + c(j)*p(t,j)^2)) + startup(j) - startup(j)*u(t-
1,j)-lamdual(t)*p(t,j)-mudual(t)*pmax(j) < 0
        u(t,j) = 1;
    else
        u(t,j) = 0;
    end;



        Minimum Up/Down Time Constraints

        Comments: Use the X(t,j) variable to compare against
        unit minimum up/down times. If constraint is not
        satisfied, the unit commitment must be the same as the
        previous hour until minimum up/down time is satisfied

        Since this is NOT the first hour, we know the units
        have been either online or offline from the previous
        hour.
        Using that information u(t-1,j) and the constraints of
        minimum up/down time in X(t,j)

        Case u(t,j) = 1 And u(t-1,j) = 0

    if u(t,j) == 1 && u(t-1,j) == 0
        if X(t,j) > minoff(j)
                Since the minimumum off time is violated, keep
                the unit offline!
            u(t,j)= 0;
                Increment the unit offtime and update X(t,j)
            X(t+1,j) = max(X(t,j)-1,minoff(j));
        else
                Increment the unit ontime and update X(t,j)
            X(t+1,j) = min(X(t,j)+1,minon(j));
        end
    else
     end


      Case u(t,j) = 1 And u(t-1,j) = 1

    if u(t,j) == 1 && u(t-1,j) == 1
          Increment the unit ontime and update X(t,j)
        X(t+1,j) = min(X(t,j)+1,minon(j));
    else
    end
```

```
            Case u(t,j) = 0 and u(t-1,j) = 1

            if u(t,j) == 0 && u(t-1,j) == 1
                if X(t,j) < minon(j)
                   Since the minimum on time is violated, keep the unit
                   online
                 u(t,j) = 1;
                   Increment the unit ontime and update X(t,j)
                 X(t+1,j) = min(X(t,j)+1,minon(j));
                else
                   Increment the unit offtime and update X(t,j)
                 X(t+1,j) = max(X(t,j)-1,minoff(j));
                end
            else
            end



            Case u(t,j) = 0 And u(t-1,j) = 0

            if u(t,j) == 0 && u(t-1,j) == 0
                  Increment the unit offtime and update X(t,j)
                X(t+1,j) = max(X(t,j)-1,minoff(j));
            else
            end

        end

    end;

end;
  --------------------------------------------------------------------------

  Update value of P's
for t=1:4
    for j=1:3
        p(t,j) = p(t,j)*u(t,j);
    end
end

  --------------------------------------------------------------------------
  Determine the Dual Solution

dual=0;

for t=1:4

    sumcost=0;
    sumpower(t)=0;
    sumpowermax(t)=0;

    for j=1:3
        if t==1
            sumcost = sumcost + (a(j) + b(j)*p(t,j) + c(j)*p(t,j)^2)*u(t,j) +
startup(j)*u(t,j);
            sumpower(t) = sumpower(t) + p(t,j)*u(t,j);
            sumpowermax(t) = sumpowermax(t) + pmax(j)*u(t,j);
        else
            sumcost = sumcost + (a(j) + b(j)*p(t,j) + c(j)*p(t,j)^2)*u(t,j) +
startup(j)*u(t,j)*(1-u(t-1,j));
            sumpower(t) = sumpower(t) + p(t,j)*u(t,j);
            sumpowermax(t) = sumpowermax(t) + pmax(j)*u(t,j);
        end
    end

      This function uses the dual lamda, mu, and Pi and unit commitment
      values of the units to determine the total value of the objective
      or dual solution
    dual = dual + sumcost + lamdual(t)*(pload(t)-sumpower(t)) + mudual(t)*(pload(t) +
reserve(t) - sumpowermax(t));
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dual(i) = dual;

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Capacity and lamda's

 % Comments: First determine the min and max capacity for each hour
 % and the min and max lambda for each hour

 for t=1:4

   pcapmin(t)=0;
   lamsmin(t)=10000;
   pcapmax(t)=0;
   lamsmax(t)=-10000;

   for j=1:3

       if u(t,j) == 1
           pcapmin(t) = pcapmin(t) + pmin(j);
           pcapmax(t) = pcapmax(t) + pmax(j);

           if lamsmin(t) > lammin(j)
               lamsmin(t) = lammin(j);
           end
           if lamsmax(t) < lammax(j)
               lamsmax(t) = lammax(j);
           end
       end
   end
   % test hourly capacity
   capok(t) = 1;
   if pcapmin(t) > pload(t)
       capok(t) = 0;
   end
   if pcapmax(t) < pload(t) + reserve(t)
       capok(t) = 0;
   end
 end

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Primal Solution

  % Comments: the primal solution for each hour will only be determine if there is
  % enough capacity in the system for that hour to meet both load and
  % reserve

 primaltot=0;

 for t=1:4

     if capok(t) == 1

         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         % Economic Dispatch

         % Comments: I know this is not the most efficient determination
         % of economic dispatch but due to time constraints, I left it
         % similar to homework example. It works!

         % lambda for primal solution
       lamedc = (lamsmin(t) + lamsmax(t))/2;
         % this is the value to change lambda
       dellam = 0.001;
       for s=1:100000
           pedctot = 0;
           for j=1:3
               if u(t,j) == 1
                   pedc(t,j) = (lamedc - b(j))/(2*c(j));
                   if pedc(t,j) < pmin(j)
                       pedc(t,j) = pmin(j);
```

```
                    end
                    if pedc(t,j) > pmax(j)
                        pedc(t,j) = pmax(j);
                    end;
                else
                    pedc(t,j)=0;
                end;
                    pedctot = pedctot + pedc(t,j);
            end
              update the lambda
            if pedctot > pload(t)
                lamedc =  lamedc -  dellam;
            end
            if pedctot < pload(t)
                lamedc = lamedc + dellam;
            end
        end

          Determine the primal solution given unit commitment, start-up
          costs, etc..
        primal(t)=0;
        for j=1:3
            if t==1
            primal(t) = primal(t) + (a(j) + b(j)*pedc(t,j) + c(j)*pedc(t,j)^2)*u(t,j)
+ startup(j)*u(t,j);
            else
            primal(t) = primal(t) + (a(j) + b(j)*pedc(t,j) + c(j)*pedc(t,j)^2)*u(t,j)
+ startup(j)*u(t,j)*(1-u(t-1,1));
            end
        end

     else

          when capacity test fails, set primal to a large number
        for j=1:3
            pedc(t,j)=0;
        end
        primal(t) = 10000000;
     end
     primaltot = primaltot + primal(t);
  end
  pri(i) = primaltot;
  ------------------------------------------------------------------------


   Duality Gap

  if dual(i) > 0
  gap(i) = abs((pri(i)-dual(i))/dual(i));
  else
      gap(i)=50000;
  end;



  ------------------------------------------------------------------------
   Update lamda and mu constraints

   Comments: Calculate the gradient of the dual function

   Updates the lamda
  for t=1:4
      graddual(t) = pload(t) - sumpower(t);
      if graddual(t) > 0
          lamdual(t) = lamdual(t) + 0.05*graddual(t);
      else
          lamdual(t) = lamdual(t) + 0.01*graddual(t);
      end
  end
```

```
    %Updates the mu
    for t=1:4
         graddual(t) = pload(t) + reserve(t) - sumpowermax(t);
         if graddual(t) > 0
             mudual(t) = mudual(t) + 0.05*graddual(t);
         else
             mudual(t) = mudual(t) + 0.01*graddual(t);
         end
    end

   if gap(i) < 0.1
     iteration = i
     dualitygap = gap(i)
     dualP = p
     primalP = pedc
     units = u
     dualf = dual(i)
     primalf = pri(i)
   else
   end

   dualgap = [dualgap; i gap(i)];

 end
```

Problem #2

Assume the fuel consumption functions of the three units are given by:

Table- Piecewise Linear Fuel Consumption Curves

Unit 1

P (MW)	H (MBtu/h)
100	660
300	2260
500	4260
600	5410

Unit 2

P (MW)	H (MBtu/h)
100	845
200	1720
300	2645
400	3620

Unit 3

P (MW)	H (MBtu/h)
50	485
100	950
150	1425
200	1910

Solve the unit commitment problem by Lagrange Relation method and iterate until two distinct feasible primal solutions are obtained. Show duality gap in each iteration. Assume $\alpha_\lambda^+ = 0.05, \alpha_\mu^+ = 0.05, \alpha_\lambda^- = 0.01, \alpha_\mu^- = 0.01$.

Problem Formulation

See tables 1 and 2 for unit parameters and system load and reserve requirements

Assumptions: Problem does not minimize CO_2 emissions, only costs. Problem DOES consider minimum up/down time constraints, unit power limits, and reserve requirements.

The formulation for Lagrange Relaxation will be similar to that of problem 1 and for brevity, please reference Problem 1 formulation for Lagrange Relaxation. However, the unit parameters and heat rates are different. Since the heat rates are linear, the calculation for P_i for each unit in both the dual and primal problems is a bit different and IMOP a little complicated. I shall discuss some algorithms and strategies that I have used in my Matlab program to tackle the linear curves and economic dispatch.

The first step was to determine the incremental costs of each unit for the piecewise linear segments. The table blow shows lambda as the piece-wise incremental cost.

Piecewise Incremental Costs

Unit 1

P (MW)	H (MBtu/h)	Slope dH/dP	Lambda
100	660	8	40
300	2260	10	50
500	4260	11.5	57.5
600	5410		

Unit 2

P (MW)	H (MBTu/h)	Slope dH/dP	Lambda
100	845	8.75	43.75
200	1720	9.25	46.25
300	2645	9.75	48.75
400	3620		

Unit 3

P (MW)	H (MBtu/h)	Slope dH/dP	Lambda
50	485	9.3	46.5
100	950	9.5	47.5
150	1425	9.7	48.5
200	1910		

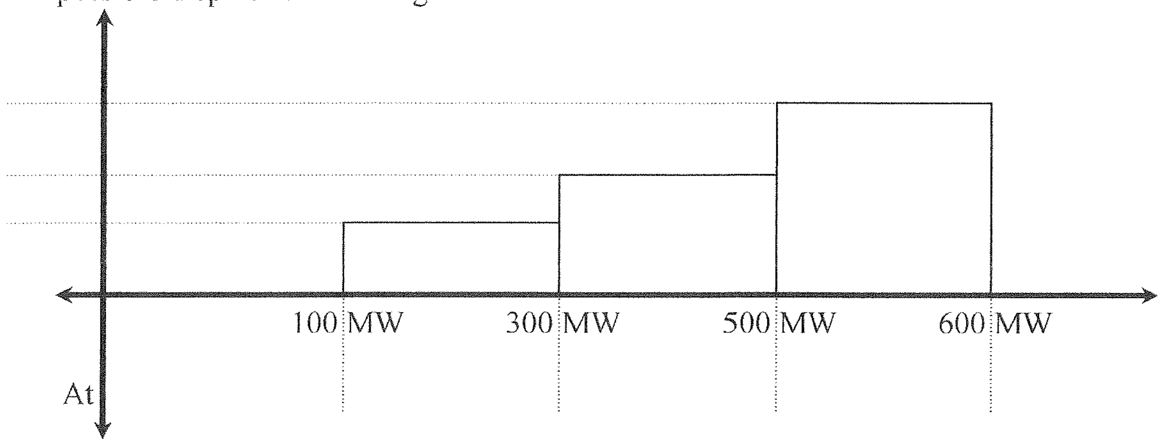
$$\lambda = (dH/dP)*(\$5/Mbtu)$$

To determine economic dispatch, the piecewise segments must be ordered by least incremental cost, or lambda. The following table shows the ordering of the units for economic dispatch, which is used by the primal problem.

Priority	Unit	Mwmin	Mwmax	Price ($/h)	Slope (dH/dP)	Hmin	Ploadmin	Ploadmax
1	1	100	300	40	8	660	100	300
2	2	100	200	43.75	8.75	845	200	400
3	2	200	300	46.25	9.25	1720	300	500
4	3	50	100	46.5	9.3	485	250	550
5	3	100	150	47.5	9.5	950	150	600
6	3	150	200	48.5	9.7	1425	250	650
7	2	300	400	48.75	9.75	2645	450	750
8	1	300	500	50	10	2260	600	950
9	1	500	600	57.5	11.5	4260	800	1050

Dual Problem

For the dual problem, the calculation of the dual P_i was a challenge. Using the λ_t value as a reference, multiple if-else statements placed the dual P_i at either P_i^{min} , P_i^{max}, or on one of the curve points for that unit. For example, if $\lambda_t = 42$, then P_i would be set to 300 MW because it's incremental cost points are either 40, 50, or 57.5. If $\lambda_t = 40$, P_i could be dispatched to any value on that range (100-300 MW), but since our objective is to find a solution, and one that satisfies capacity constraints, we set P_i to be the maximum possible dispatch at that range.



If λ_t = any λ on the piecewise curve, set that value to the maximum MW dispatch in that interval
If λ_t < smallest λ on the piecewise curve, set that value to P_{MIN}
If λ_t >= largest λ on the piecewise curve, set that value to P_{MAX}
If $\lambda_t > \lambda_c$ and $\lambda_t < \lambda_c$ on the piecewise curve, set that value to P_{MAX} at λ_c

Costs

The costs associated will be the total heat MBtu/h multiplied by the fuel cost $/MBtu. In the last problem we used the unit parameters to determine the total heat produced for the hour and the associated costs. With piecewise linear functions, the costs depend on in which segment that is being dispatched. In the dual problem, it was easy to determine the H (MBtu/h) given the determined MW for the data table. When one chooses a curve point for the dual P values, one is also choosing a curve point for the H. However in the primal problem one has to use the equation of the line to interpolate the value of H and its associated cost.

Minimum up/down time constraints were not modified per the first problem.

Start-up costs were not modified per the first problem.

Primal Problem

The economic dispatch requires a priority list of units to be dispatched by lowest incremental cost. Such a table has been made in the problem formulation and is repeated here.

Priority	Unit	Mwmin	Mwmax	Price ($/h)	Slope (dH/dP)	Hmin	Ploadmin	Ploadmax
1	1	100	300	40	8	660	100	300
2	2	100	200	43.75	8.75	845	200	400
3	2	200	300	46.25	9.25	1720	300	500
4	3	50	100	46.5	9.3	485	250	550
5	3	100	150	47.5	9.5	950	150	600
6	3	150	200	48.5	9.7	1425	250	650
7	2	300	400	48.75	9.75	2645	450	750
8	1	300	500	50	10	2260	600	950
9	1	500	600	57.5	11.5	4260	800	1050

Given a feasible unit commitment solution that meets both load and reserve, the economic dispatch of that solution for the hour can be assigned based on the priority table. First the units are dispatched per their P_{MIN} IF they are committed. A temporary variable will hold the total dispatch progress for the iteration in the economic dispatch loop. For example, the dispatch when considering priority 5 will use the dispatch from priorities 1,2,3 and 4 summed to a "load filled" variable. A high-level flow diagram of the algorithm is illustrated on the next page.

For t=1:4 hours of disoatch

Is the Dual Solution Feasible? $P_{totalMAX} >= P_{load}^{t} + Reserve^{t}$

YES

Set "loadfilled" = 0 - Sum of total power

For j=1:3

Set Minimum Dispatch For All Committed Units
Add Minimum Dispatch to "loadfilled"
Determine Costs at Minimum Dispatch.
Add to Total Cost

For w=1:9 (total priority list segments)

Is loadfilled < $Pload_{t}$

YES

Is unit at priority list # w committed?

YES

IS `(pload(t)-loadfilled) > (priMWmax(w) - priMWmin(w))`

YES

```
P(t,unit@w)=P(t,unit@w)+(MWmax(w)- MWmin(w))
loadfilled = loadfilled+ (MWmax(w)- MWmin(w))
cost(t,unit@w) = cost(t,unit@w)+ price*((MWmax(w)-
MWmin(w))*slope(w) + Hmin(w));
```

NO

```
P(t,unit@w)=P(t,unit@w)+(pload(t)- loadfilled)
cost(t,unit@w) = cost(t,unit@w)+ price*(pload(t)-
loadfilled)*slope(w) + Hmin(w));
loadfilled = loadfilled+ (pload(t)-loadfilled)
```

Results

dualgap =

Iteration	Gap
1	50000
2	6.0014
3	2.8826
4	1.811
5	1.2597
6	0.96271
7	0.76742
8	203.12
9	0.8307
10	0.569
11	94.124
12	88.908
13	83.894
14	73.514
15	0.28816
16	77.095
17	361.17
18	82.092
19	0.16437
20	331.97
21	0.19942
22	126.08
23	0.4396
24	0.27002
25	0.14044
26	0.039446
27	0.047953
28	354.47
29	0.092769
30	64
31	61.748
32	58.66
33	0.13535
34	348.88
35	57.63
36	46.506
37	0.17065
38	50.683
39	378.74
40	54.996
41	51.749
42	340.27
43	51.509
44	0.36417
45	79.862
46	0.13147
47	0.19622
48	0.25162
49	0.29753
50	0.33705

dualgap =

Iteration	Gap
51	359.26
52	223.29
53	0.25116
54	0.30995
55	42.34
56	345.71
57	42.776
58	67.847
59	0.25886
60	0.30897
61	0.35197
62	0.38871
63	0.42129
64	0.39612
65	50000
66	0.17693
67	0.25148
68	0.31178
69	0.35927
70	0.39694
71	36.997
72	58.941
73	0.43159
74	35.089
75	364.94
76	35.651
77	0.46694
78	50000
79	0.2521
80	0.31416
81	0.36605
82	0.4087
83	0.4417
84	34.671
85	34.355
86	34.017
87	33.071
88	32.175
89	31.325
90	0.52352
91	50000
92	0.35938
93	0.40535
94	0.44303
95	0.47482
96	0.49999
97	31.058
98	30.689
99	30.548
100	29.778

Solution # 1

iteration =

26

dualitygap =

0.0394

dualP =

0	400	200
600	400	200
500	400	200
300	100	0

primalP =

0	250	50
300	250	50
300	300	200
300	100	0

units =

0	1	1
1	1	1
1	1	1
1	1	0

dualf =

1.4647e+005

primalf =

152250

Solution # 2

iteration =

29

dualitygap =

0.0928

dualP =

0	300	0
600	400	200
600	400	200
600	400	0

primalP =

0	300	0
300	250	50
300	300	200
300	100	0

units =

0	1	0
1	1	1
1	1	1
1	1	0

dualf =

1.3922e+005

primalf =

1.5214e+005

```
  ECE 564   Spring 2007
  Exam # [illegible]  Midterm
  Due March [illegible], 2007
  Kathleen E. Williams

  Problem #2 Unit Commitment with Piecewise Linear Fuel Consumption
 ----------------------------------------------------------------------

echo off
clear

  Fuel Price $/MBtu
price = 5;

  Piecewise Linear Fuel Consumption Curve
MW1 = [100, 100, 50];
MW2 = [300, 200, 100];
MW3 = [500, 300, 150];
MW4 = [600, 400, 200];
H1 = [660, 845, 485];
H2 = [2260, 1720, 950];
H3 = [4260, 2645, 1425];
H4 = [5410, 3620, 1910];

  determines lambda search range for economic dispatch in primal problem
    for j=1:3
    lammin(j) = price*[(H2(j)-H1(j))/(MW2(j)-MW1(j))];
    lammax(j) = price*[(H4(j)-H3(j))/(MW4(j)-MW3(j))];
    end

  hourly load and reserve
pload=[300,600,800,400];
reserve=[15,30,40,20];

  units min, max values
pmin=[100,100,50];
pmax=[600,400,200];

  startup costs of units
startup=[1000,500,0];

  min down-time, up time, and initial conditions of units
minon=[3,2,1];
minoff=[-3,-2,-1];
init=[-2,-3,-2];

  initializes the lambda and mu dual variables for the lagrange relaxation
lamdual=[0,0,0,0];
mudual=[0,0,0,0];

  keeps track the duality gap in each iteration
dualgap = [];

 -----------------------------------------------------------------------
  Lagrange Relaxation Loop

  Comments: Instead of stopping the loop by the value of the duality gap
  I chose to iterate multiple times to find multiple gaps and hence,
  multiple unit commitment solutions
 -----------------------------------------------------------------------

for i=1:100


  Initialize Z

  Comments: Z holds the unit up/downtime. For example, Z(t,j) means the
  min up/down time at hour t and unit j. If Z(t,j) were equal to 2, that
  means the unit has been online for [illegible] hours. If Z(t,j) were equal to [illegible],
  that means the unit has been offline for two hours
  t = hour
```

```
  p = init

for t=1:4
   for j=1:3
       X(t,j) = 0;
       if t == 1
           X(t,j) = init(j);    takes initial conditions for hour 1
       else
       end
   end
end


     ---------------------------------------------------------------------
      Dual Loop

      This begins the dual function calculation

     for t=1:4

         for j=1:3


              Determine which slope range lambda falls into

           if lamdual(t) >= price*[(H4(j)-H3(j))/(MW4(j)-MW3(j))]
                p(t,j) = MW4(j);
                h(t,j) = H4(j);
           else
                if lamdual(t) < price*[(H2(j)-H1(j))/(MW2(j)-MW1(j))];
                    p(t,j) = MW1(j);
                    h(t,j) = H1(j);
                else
                    if lamdual(t) < price*[(H3(j)-H2(j))/(MW3(j)-MW2(j))]
                        p(t,j) = MW2(j);
                        h(t,j) = H2(j);
                    else
                        if lamdual(t) < price*[(H4(j)-H3(j))/(MW4(j)-MW3(j))]
                            p(t,j) = MW3(j);
                            h(t,j) = H3(j);
                        else
                            p(t,j) = MW1(j);
                            h(t,j) = H1(j);
                        end
                    end
                end
           end
        end


        for j=1:3

              Special Condition is at Hour 1
           if t==1

                  Unit Commitment

                  Comments: Determine commitment of unit j at hour t given
                  dual values of lambda and mu at minimal

                if h(t,j)*price + startup(j) -lamdual(t)*p(t,j)-mudual(t)*pmax(j) < 0
                    u(t,j) = 1;
                else
                    u(t,j) = 0;
                end;



                  Minimum Up/Down Time Constraints
```

```
        Comments: Use the X(t,j) variable to compare against
        unit minimum up/down times. If constraint is not
        satisfied, the unit commitment must be the same as the
        previous hour until minimum up/down time is satisfied

        Since this is the first hour, we know the units are
        initially offline, so we need only consider two
        possibilities: turning the unit on, or off
      if u(t,j) == 1
          if X(t,j) > minoff(j)
                Since the minimumum off time is violated, keep
                the unit offline!
              u(t,j)= 0;
                Increment the unit offtime and update X(t,j)
              X(t+1,j) = max(X(t,j)-1,minoff(j));
          else
                The unit does NOT violate minimum off time, keep
                the unit committed and increment the unit ontime
                and update X(t,j)
              X(t+1,j) = min(X(t,j)+1,minon(j));
          end

      else
            The unit j is not committed, so we only need to
            increment the unit offtime and update X(t,j)
          X(t+1,j) = max(X(t,j)-1,minoff(j));
      end



    Considering all other hours t > 1

  else

        Unit Commitment

        Comments: Determine commitment of unit j at hour t given
        dual values of lambda and mu at minimal

      if h(t,j)*price + startup(j) - startup(j)*u(t-1,j)-lamdual(t)*p(t,j)-
mudual(t)*pmax(j) < 0
          u(t,j) = 1;
      else
          u(t,j) = 0;
      end;



        Minimum Up/Down Time Constraints

        Comments: Use the X(t,j) variable to compare against
        unit minimum up/down times. If constraint is not
        satisfied, the unit commitment must be the same as the
        previous hour until minimum up/down time is satisfied

        Since this is NOT the first hour, we know the units
        have been either online or offline from the previous
        hour.
        Using that information u(t-1,j) and the constraints of
        minimum up/down time in X(t,j)

        Case u(t,j) = 1 and u(t-1,j) = 0

      if u(t,j) == 1 && u(t-1,j) == 0
```

```
                if X(t,j) > minoff(j)
                      Since the minimumum off time is violated, keep
                      the unit offline!
                    u(t,j)= 0;
                      Increment the unit offtime and update X(t,j)
                    X(t+1,j) = max(X(t,j)-1,minoff(j));
                else
                      Increment the unit ontime and update X(t,j)
                    X(t+1,j) = min(X(t,j)+1,minon(j));
                end
           else
            end


             Case u(t,j) = 1 And u(t-1,j) = 1

           if u(t,j) == 1 && u(t-1,j) == 1
                 Increment the unit ontime and update X(t,j)
                X(t+1,j) = min(X(t,j)+1,minon(j));
           else
           end


             Case u(t,j) = 0 u(t-1,j) = 1

           if u(t,j) == 0 && u(t-1,j) == 1
               if X(t,j) < minon(j)
                  Since the minimum on time is violated, keep the unit
                  online!
                u(t,j) = 1;
                  Increment the unit ontime and update X(t,j)
                X(t+1,j) = min(X(t,j)+1,minon(j));
               else
                  Increment the unit offtime and update X(t,j)
                X(t+1,j) = max(X(t,j)-1,minoff(j));
               end
           else
           end


             Case u(t,j) = 0 And u(t-1,j) = 0

           if u(t,j) == 0 && u(t-1,j) == 0
                 Increment the unit offtime and update X(t,j)
               X(t+1,j) = max(X(t,j)-1,minoff(j));
           else
           end

        end

    end;

end;
---------------------------------------------------------------------------

  Update p and h
for t=1:4
    for j=1:3
        p(t,j) = p(t,j)*u(t,j);
        h(t,j) = h(t,j)*u(t,j);
    end
end


---------------------------------------------------------------------------
  Determine the Dual Solution

dual=0;
```

```
    for t=1:4

        sumcost=0;
        sumpower(t)=0;
        sumpowermax(t)=0;

        for j=1:3
            if t==1
                sumcost = sumcost + h(t,j)*price*u(t,j) + startup(j)*u(t,j);
                sumpower(t) = sumpower(t) + p(t,j)*u(t,j);
                sumpowermax(t) = sumpowermax(t) + pmax(j)*u(t,j);
            else
                sumcost = sumcost + h(t,j)*price*u(t,j) + startup(j)*u(t,j)*(1-u(t-1,j));
                sumpower(t) = sumpower(t) + p(t,j)*u(t,j);
                sumpowermax(t) = sumpowermax(t) + pmax(j)*u(t,j);
            end
        end

          This function uses the dual lamda, mu, and Pi and unit commitment
          values of the units to determine the total value of the objective
          or dual solution
        dual = dual + sumcost + lamdual(t)*(pload(t)-sumpower(t)) + mudual(t)*(pload(t) +
reserve(t) - sumpowermax(t));
    end
     --------------------------------------------------------------------------
    dual(i) = dual;


     --------------------------------------------------------------------------
      Capacity and lamda's

      Comments: First determine the min and max capacity for each hour
      and the min and max lambda for each hour

     for t=1:4

       pcapmin(t)=0;
       pcapmax(t)=0;

       for j=1:3

           if u(t,j) == 1
               pcapmin(t) = pcapmin(t) + pmin(j);
               pcapmax(t) = pcapmax(t) + pmax(j);
           end
       end
         test hourly capacity
       capok(t) = 1;
       if pcapmin(t) > pload(t)
           capok(t) = 0;
       end
       if pcapmax(t) < pload(t) + reserve(t)
           capok(t) = 0;
       end
     end

      ------------------------------------------------------------------------------
       Primal Solution

       Comments: The primal solution for each hour will only be determine if there is
       enough capacity in the system for that hour to meet both load and
       reserve

     primaltot=0;
     priunits = [1,2,2,3,3,3,2,1,1];
     prilambda = [40,43.75,46.25,46.5,47.5,48.5,48.75,50,57.5];
     priMWmin = [100, 100, 200, 50, 100, 150, 300, 300, 500];
     priMWmax = [300, 200, 300, 100, 150, 200, 400, 500, 600];
     slope = [8, 8.75, 9.25, 9.3, 9.5, 9.7, 9.75, 10, 11.5];
     Hmin = [660, 845, 1720, 485, 950, 1425, 2645, 2260, 4260];
```

```
 loadMax = [300, 400, 500, 550, 600, 650, 750, 950, 1050];

for t=1:4
   for j=1:3
     pedc(t,j) = 0;
   end
end

 for t=1:4

     if capok(t) == 1

         ------------------------------------------------------------------
          Economic Dispatch


          use priority table for dispatching the rest of the load




        loadfilled = 0;

          Fill all units to minimum capacity first
         for j=1:3
            pedc(t,j) = pmin(j)*u(t,j);
            loadfilled = loadfilled + pedc(t,j);
            cost(t,j) = H1(j)*price*u(t,j);
         end

        for w=1:9

            if loadfilled < pload(t)
              if u(t,priunits(w)) == 1
                  if (pload(t)-loadfilled) > (priMWmax(w) - priMWmin(w))
                        Fill the maximum MW range
                      pedc(t,priunits(w)) = pedc(t,priunits(w)) +  (priMWmax(w) -
priMWmin(w));

                      loadfilled = loadfilled + (priMWmax(w) - priMWmin(w));
                        add the filling to the filled load
                        cost
                      cost(t,priunits(w)) = cost(t,priunits(w)) + price*((priMWmax(w)-
priMWmin(w))*slope(w) + Hmin(w));
                  else
                      pedc(t,priunits(w)) = pedc(t,priunits(w)) +  (pload(t)-
loadfilled);

                      cost(t,priunits(w)) = cost(t,priunits(w)) + price*((pload(t)-
loadfilled)*slope(w) + Hmin(w));
                      loadfilled = loadfilled + (pload(t)-loadfilled);
                  end

              else
              end
            else
                  do nothing
            end

            w = 1+1;
        end




          Detemine the primal solution given unit commitment, start up
          costs, etc..
         primal(t)=0;
         for j=1:3
             if t==1
             primal(t) = primal(t) + cost(t,j)*u(t,j) + startup(j)*u(t,j);
             else
```

```
                primal(t) = primal(t) + cost(t,j)*u(t,j) + startup(j)*u(t,j)*(1-u(t-1,1));
                end
            end

          else

              primal(t) = 10000000;

          end

        primaltot = primaltot + primal(t);
    end
    pri(i) = primaltot;



      Duality Gap

    if dual(i) > 0
    gap(i) = abs((pri(i)-dual(i))/dual(i));
    else
        gap(i)=50000;
    end;



      Update lamda and mu constraints

      Comments: Calculate the gradient of the dual function

      Updates the lamda
    for t=1:4
        graddual(t) = pload(t) - sumpower(t);
        if graddual(t) > 0
            lamdual(t) = lamdual(t) + 0.05*graddual(t);
        else
            lamdual(t) = lamdual(t) + 0.01*graddual(t);
        end
    end

     Updates the mu
   for t=1:4
        graddual(t) = pload(t) + reserve(t) - sumpowermax(t);
        if graddual(t) > 0
            mudual(t) = mudual(t) + 0.05*graddual(t);
        else
            mudual(t) = mudual(t) + 0.01*graddual(t);
        end
   end

  if gap(i) < 0.1
    iteration = i
    dualitygap = gap(i)
    dualP = p
    primalP = pedc
    units = u
    dualf = dual(i)
    primalf = pri(i)
  else
  end

  dualgap = [dualgap; i gap(i)];

end
```

Problem #3

Give the unit commitment in table 4, solve the economic dispatch considering ramping constraints and compare the solutions to those without ramping constraints.

Given Unit Commitment and Economic Dispatch

Economic Dispatch No Ramping Constraints

Hour	U1	U2	U3	P1	P2	P3
1	0	1	0	0	300	0
2	1	1	0	333.34	266.68	0
3	1	1	1	354.54	309.08	136.35
4	1	1	0	266.66	133.32	0

$$P_i^{MIN}{}_{(t+1)} = \min(P_i^{MIN}, P_{it}^{ED}\text{-ramprate})$$
$$P_i^{MAX}{}_{(t+1)} = \min(P_i^{MAX}, P_{it}^{ED}\text{+ramprate})$$

The initial conditions given in the unit parameter table show that unit two has been offline for initially 3 hours. To satisfy the ramping constraints for this problem, this would mean that unit 2 should be online at hour = 0 at its minimum of 100 MW to be able to dispatch 300 MW in hour one. Given the minimum up/down time constraints, this would be feasible. Given the problem initial conditions, and economic dispatch, the original solution would not satisfy ramping constraints unless hour = 0 initial conditions are changed.

Hour # 1 Start Consideration

Economic Dispatch Ramping Starts at Hour 1

Hour	U1	U2	U3	P1	P2	P3
1	0	1	0	0	300	0
2	1	1	0	250	350	0
3	1	1	1	366.66	333.32	100
4	1	1	0	216.67	183.32	0
Hour	**P1MAX**	**P2MAX**	**P3MAX**			
1	X	300	X			
2	250	400	X			
3	500	400	100			
4	600	400	X			
Hour	**P1MIN**	**P1MIN**	**P1MIN**			
1	X	300	X			
2	100	150	X			
3	100	200	50			
4	116.66	183.32	X			

In Hour # 1, the economic dispatch is taken from the original problem without ramping constraints. The P_{MIN} and P_{MAX} values for that hour are unaffected and hence the dispatch is the same. I will note that the Hour =0 value for Unit 2 should be at minimum 100 MW to account for ramping of unit 2 to meet the economic dispatch of 300 MW. Since the initial conditions given in the unit parameters show unit 2 as being offline for 3 hours before hour 1, it would seem that one can not start the unit at hour 1 above its maximum ramping constraint of 250 MW.

For Hour 2, the P_{MIN} and P_{MAX} will be affected by the ramping constraints and economic dispatch of hour one. The table above shows the iteration. First determine the economic dispatch of the starting hour, then update the P_{MIN} and P_{MAX} values of neighboring hours 0-2. Then perform another economic dispatch on those neighboring hours. Hour 2 P_{MIN} and P_{MAX} values were updated after Hour 1 economic dispatch, then an economic dispatch was done based on the new P_{MIN} and P_{MAX} values. Hour 3 P_{MIN} and P_{MAX} values were updated and so on.

Hour # 2 Start Consideration

Using the same algorithm described above, the economic dispatch and P_{MIN} and P_{MAX} values are determined.

Economic Dispatch Ramping Starts at Hour 2

Hour	U1	U2	U3	P1	P2	P3
1	0	1	0	0	300	0
2	1	1	0	333.34	266.68	0
3	1	1	1	366.68	333.36	100
4	1	1	0	216.64	183.36	0
Hour	**P1MAX**	**P2MAX**	**P3MAX**			
1	X	400	X			
2	333.34	266.68	X			
3	583.34	400	100			
4	600	400	X			
Hour	**P1MIN**	**P1MIN**	**P1MIN**			
1	83.34	116.68	X			
2	333.34	266.68	X			
3	83.34	116.68	50			
4	116.68	183.36	X			

Note that given the economic dispatch at hour two, it is infeasible to ramp up unit one because the economic dispatch at hour 2 exceeds its ramp rate and the unit one is not committed at hour one. One would have to commit the unit at hour one to its minimum dispatch to be able to satisfy the ramping requirements for hour 2. However, this is impossible because it would violate the minimum downtime constraints for this unit.

Hour # 3 Start Consideration

Economic Dispatch Ramping Starts at Hour 3

Hour	U1	U2	U3	P1	P2	P3
1	0	1	0	0	300	0
2	1	1	0	333.33	266.66	0
3	1	1	1	354.54	309.08	136.35
4	1	1	0	240.93	159.08	0
Hour	**P1MAX**	**P2MAX**	**P3MAX**			
1	X	400	X			
2	600	400	X			
3	354.54	309.08	136.35			
4	600	400	X			
Hour	**P1MIN**	**P1MIN**	**P1MIN**			
1	83.33	116.66	X			
2	104.54	159.08	36.35			
3	354.54	309.08	136.35			
4	104.54	159.08	36.35			

In hour three starting, the same algorithm of updating the neighbors P_{MIN} and P_{MAX} , economic dispatch, and updates are repeated. It is interesting that the same of problem seen when starting at hour two is seen here with unit 1. At hour 1, unit 1 must be committed to account for ramping constraints, however, this is against the unit commitment solution. Similarly, for unit three, in order to satisfy the ramping constraint at hour three, the unit must be committed at hours two and four, which is also against the unit commitment solution.

Hour #4 Start Consideration

Economic Dispatch Ramping Starts at Hour 4

Hour	U1	U2	U3	P1	P2	P3
1	0	1	0	0	300	0
2	1	1	0	333.34	266.68	0
3	1	1	1	416.67	283.32	100
4	1	1	0	266.66	133.32	0
Hour	**P1MAX**	**P2MAX**	**P3MAX**			
1	X	400	X			
2	600	400	X			
3	516.66	283.32	100			
4	266.66	133.32	X			
Hour	**P1MIN**	**P1MIN**	**P1MIN**			
1	X	100	X			
2	166.67	133.32	X			
3	100	100	50			
4	266.66	133.32	X			

In hour three starting, the same algorithm of updating the neighbors P_{MIN} and P_{MAX} , economic dispatch, and updates are repeated. It is interesting that the same of problem seen when starting at hour two and hour three is seen here with unit 1. At hour 1, unit 1 must be committed to account for ramping constraints, however, this is against the unit commitment solution.

The best solution would be to start at hour # 1. With the minimum up/down time constraints, this would be feasible; however given the problem initial conditions, and economic dispatch, the original solution would not satisfy ramping constraints unless hour = 0 initial conditions are changed. Unit two at hour = 0 would have to be set at its minimum of 100 MW to allow 300 MW dispatch at hour one.

```
  ----------------------------------------------------------------
  ECE 564   Spring 2007
  Exam # 1 -Midterm
  Due March 12, 2007
  Kathleen R. Williams

  Problem #3 Unit Commitment with Quadratic Fuel Consumption Function
  ----------------------------------------------------------------

echo off
clear

  fuel consumption parameters times the price ($1.0 /MBTU)
a=[10,20,30]*5;
b=[6,8,9]*5;
c=[0.005,0.0025,0.002]*5;

  hourly load and reserve
pload=[300,600,800,400];
reserve=[15,30,40,20];

  units min, max values
  these values are changed based on economic dispatch
  and starting hour conditions
pmin=[166.67,133.32,50];
pmax=[600,400,100];

  startup costs of units
startup=[1000,500,0];

  min down time, up time, and initial conditions of units
minon=[3,2,1];
minoff=[-3,-2,-1];
init=[-2,-3,-2];

  determines lambda search range for economic dispatch in primal problem
for j=1:3
    lammin(j) = b(j)+ 2*c(j)*pmin(j);
    lammax(j) = b(j)+ 2*c(j)*pmax(j);
end

  the unit commitment is assumed to remain constant per the problem
  definition
u = [0 1 0; 1 1 0;1 1 1 ;1 1 0];
     for t=1:4
       pcapmin(t)=0;
       lamsmin(t)=10000;
       pcapmax(t)=0;
       lamsmax(t)=-10000;
       for j=1:3
```

```
            if u(t,j) == 1
                pcapmin(t) = pcapmin(t) + pmin(j);
                pcapmax(t) = pcapmax(t) + pmax(j);

                if lamsmin(t) > lammin(j)
                    lamsmin(t) = lammin(j);
                end
                if lamsmax(t) < lammax(j)
                    lamsmax(t) = lammax(j);
                end
            end
        end
          test hourly capacity
        capok(t) = 1;
        if pcapmin(t) > pload(t)
            capok(t) = 0;
        end
        if pcapmax(t) < pload(t) + reserve(t)
            capok(t) = 0;
        end
    end
    primaltot=0;
    for t=1:4
        if capok(t) == 1
            ---------------------------------------------------------------
             Economic Dispatch

             Comments: I know this is not the most efficient determination
             of economic dispatch but due to time constraints, I left it
             similar to homework example. It works!
             Lambda for primal solution
            lamedc = (lamsmin(t) + lamsmax(t))/2;
             this is the value to change lambda
            dellam = 0.001;
            for s=1:100000
                pedctot = 0;
                for j=1:3
                    if u(t,j) == 1
                        pedc(t,j) = (lamedc - b(j))/(2*c(j));
                        if pedc(t,j) < pmin(j)
                            pedc(t,j) = pmin(j);
                        end
                        if pedc(t,j) > pmax(j)
                            pedc(t,j) = pmax(j);
                        end;
                    else
                        pedc(t,j)=0;
                    end;
                        pedctot = pedctot + pedc(t,j);
                end
                  update the lambda
                if pedctot > pload(t)
                    lamedc =  lamedc -  dellam;
                end
                if pedctot < pload(t)
                    lamedc = lamedc + dellam;
                end
            end
             Determine the primal solution given unit commitment, start up
             costs, etc.
            primal(t)=0;
            for j=1:3
                if t==1
                primal(t) = primal(t) + (a(j) + b(j)*pedc(t,j) + c(j)*pedc(t,j)^2)*u(t,j)
+ startup(j)*u(t,j);
                else
                primal(t) = primal(t) + (a(j) + b(j)*pedc(t,j) + c(j)*pedc(t,j)^2)*u(t,j)
+ startup(j)*u(t,j)*(1-u(t-1,1));
                end
            end
        else
```

```
        when capacity test fails, set primal to a large number
      for j=1:3
          pedc(t,j)=0;
      end
      primal(t) = 10000000;
   end
   primaltot = primaltot + primal(t);
end
The program performs the economic dispatch for all four hours, but
only take the row for the current hour being done
```

Problem # 4

Given emission constrains, use Lagrange relaxation on economic dispatch

The formulation is as follows:

load constraint

$$\mathcal{L} = \min \sum_{t=1}^{4}\sum_{u=1}^{3} F_i(P_{i,t})U_{i,t} + \mu\left[\sum_{t=1}^{4}\sum_{u=1}^{3} E(P_{i,t}) - 1800\right] + \lambda\left[\sum_{t}^{4} P_{load} - \sum_{i} [illegible]\right]$$

Where $E(P_{i,t})$ is the emissions of unit i at [illegible]

$\mathcal{E}(P_{i,t}) = [a + bP_{i,t} + cP_{i,t}^2] \times e_i$ where e_i is the emission rate lb/MBtu

$P_{i,t} = (-5b_i - \mu e_i b_i + \lambda)/(10a_i + [illegible])$

where a_i, b_i, c_i are unit parameters

$0 < \mu < 200$ (given in class)

Unit commitment

0	1	0
1	1	0
1	1	1
1	1	0

$P_i^{min} < P_{i,t} < P_i^{max}$

1800 = emission (tons/hour) or "allowance"

% Error = 100 × (allowance − total emissions)/allowance

An allowance of 1800 tons is not sufficient enough for this unit commitment. I was able to get results based on 1850 tons requirements. I converted all emissions to tons and varied the value of μ in steps of 1 between 1 and 200. I performed economic dispatch based on the values of μ and λ.

In class lecture it was given than μ should be between 1 and 200.

For an allowance of 1800 tons

Mu	allowance	Emission	% Percent Error
1	1800	1912.1	-6.2261
2	1800	1891.3	-5.0708
3	1800	1901.9	-5.6607
4	1800	1874.3	-4.1292
5	1800	1878.4	-4.3568
6	1800	1866.9	-3.7176
7	1800	1856	-3.1096
8	1800	1867.5	-3.7495
9	1800	1857.1	-3.1722
10	1800	1868.3	-3.7939
11	1800	1858.4	-3.244
12	1800	1869.3	-3.8481
13	1800	1859.8	-3.3227
14	1800	1850.7	-2.8192
15	1800	1862.5	-3.4709
16	1800	1855.7	-3.092
17	1800	1865	-3.6137
18	1800	1858.5	-3.2479
19	1800	1849.5	-2.7514
20	1800	1861.2	-3.3993
21	1800	1846.3	-2.5718
22	1800	1857.7	-3.2054
23	1800	1849.4	-2.7437
24	1800	1860.4	-3.3581
25	1800	1852.4	-2.9093
26	1800	1846.9	-2.6062
27	1800	1857.6	-3.199
28	1800	1849.9	-2.7739
29	1800	1860.3	-3.3497
30	1800	1852.8	-2.9354
31	1800	1847.8	-2.6568
32	1800	1855.6	-3.091
33	1800	1850.7	-2.819
34	1800	1858.3	-3.241
35	1800	1853.6	-2.9751
36	1800	1846.8	-2.6006
37	1800	1851.3	-2.8514
38	1800	1844.8	-2.4884
39	1800	1854.1	-3.0033
40	1800	1847.7	-2.6479
41	1800	1843.4	-2.4116
42	1800	1850.4	-2.8016
43	1800	1846.3	-2.5697
44	1800	1853.1	-2.9497
45	1800	1849	-2.7221
46	1800	1843.1	-2.396
47	1800	1851.6	-2.8691
48	1800	1845.9	-2.549
49	1800	1854.2	-3.0109
50	1800	1848.5	-2.6965
51	1800	1844.8	-2.4883
52	1800	1851.1	-2.839
53	1800	1843.3	-2.4071
54	1800	1849.5	-2.752
55	1800	1845.9	-2.5526
56	1800	1840.7	-2.2634
57	1800	1848.5	-2.6933
58	1800	1843.4	-2.4086
59	1800	1850.9	-2.8292
60	1800	1845.9	-2.549
61	1800	1842.6	-2.3646
62	1800	1848.3	-2.6848
63	1800	1845.1	-2.5029
64	1800	1850.7	-2.8162
65	1800	1847.5	-2.6368
66	1800	1842.7	-2.3747
67	1800	1849.8	-2.7664
68	1800	1845.1	-2.508
69	1800	1848.6	-2.6982

Mu	allowance	Emission	% Percent Error
70	1800	1844	-2.4452
71	1800	1841	-2.2796
72	1800	1846.3	-2.5738
73	1800	1843.4	-2.4101
74	1800	1839	-2.1688
75	1800	1845.7	-2.5367
76	1800	1841.4	-2.2985
77	1800	1847.9	-2.6596
78	1800	1843.6	-2.4244
79	1800	1840.9	-2.2708
80	1800	1845.8	-2.5466
81	1800	1843.1	-2.3946
82	1800	1848	-2.6653
83	1800	1845.3	-2.5149
84	1800	1841.3	-2.2924
85	1800	1844.3	-2.4629
86	1800	1840.4	-2.2444
87	1800	1846.4	-2.5791
88	1800	1842.5	-2.3631
89	1800	1840	-2.2225
90	1800	1844.6	-2.4786
91	1800	1842.1	-2.3394
92	1800	1846.6	-2.591
93	1800	1844.2	-2.4531
94	1800	1840.5	-2.2478
95	1800	1846.1	-2.5637
96	1800	1842.5	-2.3607
97	1800	1848.1	-2.6715
98	1800	1844.5	-2.4706
99	1800	1842.1	-2.3398
100	1800	1846.4	-2.5777
101	1800	1841.4	-2.2983
102	1800	1837.9	-2.1058
103	1800	1843.3	-2.4058
104	1800	1839.9	-2.2152
105	1800	1845.2	-2.5107
106	1800	1841.8	-2.322
107	1800	1839.6	-2.1996
108	1800	1843.7	-2.4261
109	1800	1841.5	-2.3046
110	1800	1845.5	-2.5276
111	1800	1843.3	-2.4072
112	1800	1840.1	-2.2267
113	1800	1845.1	-2.5073
114	1800	1841.9	-2.3284
115	1800	1846.9	-2.605
116	1800	1843.7	-2.4278
117	1800	1839.2	-2.1783
118	1800	1843	-2.3912
119	1800	1841	-2.2779
120	1800	1837.9	-2.1072
121	1800	1842.8	-2.3752
122	1800	1839.7	-2.206
123	1800	1844.5	-2.4703
124	1800	1841.4	-2.3025
125	1800	1839.5	-2.1941
126	1800	1843.1	-2.3969
127	1800	1841.2	-2.2892
128	1800	1844.8	-2.4892
129	1800	1842.9	-2.3823
130	1800	1840	-2.221
131	1800	1844.5	-2.4733
132	1800	1841.6	-2.3134
133	1800	1843.9	-2.4403
134	1800	1841.1	-2.2824
135	1800	1839.2	-2.1804
136	1800	1842.7	-2.3722
137	1800	1840.9	-2.2709
138	1800	1838.1	-2.1175

Mu	allowance	Emission	% Percent Error
139	1800	1842.5	-2.3595
140	1800	1839.7	-2.2073
141	1800	1844	-2.4462
142	1800	1841.3	-2.2952
143	1800	1839.6	-2.1977
144	1800	1842.9	-2.3814
145	1800	1841.1	-2.2845
146	1800	1844.4	-2.4658
147	1800	1842.7	-2.3695
148	1800	1840	-2.2237
149	1800	1842.1	-2.3411
150	1800	1839.5	-2.197
151	1800	1843.6	-2.4239
152	1800	1841.1	-2.2809
153	1800	1839.4	-2.1886
154	1800	1842.5	-2.3631
155	1800	1840.9	-2.2714
156	1800	1838.4	-2.132
157	1800	1842.3	-2.3526
158	1800	1839.9	-2.2142
159	1800	1843.8	-2.4322
160	1800	1841.3	-2.2948
161	1800	1839.7	-2.2062
162	1800	1842.7	-2.3739
163	1800	1841.1	-2.2858
164	1800	1844.1	-2.4516
165	1800	1840.7	-2.261
166	1800	1838.3	-2.1284
167	1800	1842.1	-2.3387
168	1800	1839.7	-2.207
169	1800	1843.5	-2.4149
170	1800	1841.1	-2.2841
171	1800	1839.6	-2.1999
172	1800	1842.5	-2.3599
173	1800	1841	-2.2761
174	1800	1838.7	-2.1484
175	1800	1842.3	-2.3509
176	1800	1840	-2.2241
177	1800	1843.6	-2.4244
178	1800	1841.4	-2.2984
179	1800	1839.9	-2.2172
180	1800	1842.7	-2.3715
181	1800	1839.5	-2.1951
182	1800	1842.3	-2.3482
183	1800	1840.8	-2.2682
184	1800	1838.6	-2.1462
185	1800	1842.1	-2.34
186	1800	1839.9	-2.2188
187	1800	1838.5	-2.1409
188	1800	1841.2	-2.2901
189	1800	1839.8	-2.2126
190	1800	1842.5	-2.3603
191	1800	1841.1	-2.2831
192	1800	1839	-2.1653
193	1800	1842.3	-2.3525
194	1800	1840.2	-2.2353
195	1800	1843.6	-2.4207
196	1800	1841.5	-2.3043
197	1800	1838.5	-2.1402
198	1800	1841.1	-2.2834
199	1800	1839.8	-2.2091
200	1800	1842.3	-2.3509

For an allowance of 1850 tons

Mu	allowance	Emission	% Percent Error
1	1850	1912.1	-3.3551
2	1850	1891.3	-2.231
3	1850	1901.9	-2.805
4	1850	1874.3	-1.3149
5	1850	1878.4	-1.5364
6	1850	1866.9	-0.91442
7	1850	1856	-0.32289
8	1850	1867.5	-0.94544
9	1850	1857.1	-0.38373
10	1850	1868.3	-0.98869
11	1850	1858.4	-0.45358
12	1850	1869.3	-1.0414
13	1850	1859.8	-0.53019
14	1850	1850.7	-0.040323
15	1850	1862.5	-0.6744
16	1850	1855.7	-0.30569
17	1850	1865	-0.81332
18	1850	1858.5	-0.45743
19	1850	1849.5	0.025692
20	1850	1861.2	-0.60471
21	1850	1846.3	0.20038
22	1850	1857.7	-0.41608
23	1850	1849.4	0.033188
24	1850	1860.4	-0.56468
25	1850	1852.4	-0.12801
26	1850	1846.9	0.16692
27	1850	1857.6	-0.40979
28	1850	1849.9	0.0037604
29	1850	1860.3	-0.55647
30	1850	1852.8	-0.15339
31	1850	1847.8	0.1177
32	1850	1855.6	-0.30479
33	1850	1850.7	-0.040061
34	1850	1858.3	-0.45071
35	1850	1853.6	-0.192
36	1850	1846.8	0.17242
37	1850	1851.3	-0.071608
38	1850	1844.8	0.28152
39	1850	1854.1	-0.21943
40	1850	1847.7	0.12633
41	1850	1843.4	0.3563
42	1850	1850.4	-0.02318
43	1850	1846.3	0.20245
44	1850	1853.1	-0.16731
45	1850	1849	0.054166
46	1850	1843.1	0.37144
47	1850	1851.6	-0.088841
48	1850	1845.9	0.22262
49	1850	1854.2	-0.22684
50	1850	1848.5	0.079033
51	1850	1844.8	0.28161
52	1850	1851.1	-0.059576
53	1850	1843.3	0.36066
54	1850	1849.5	0.025105
55	1850	1845.9	0.21905
56	1850	1840.7	0.50046
57	1850	1848.5	0.082231
58	1850	1843.4	0.35918
59	1850	1850.9	-0.050039
60	1850	1845.9	0.22259
61	1850	1842.6	0.40199
62	1850	1848.3	0.090472
63	1850	1845.1	0.26744
64	1850	1850.7	-0.037389
65	1850	1847.5	0.13721
66	1850	1842.7	0.39219
67	1850	1849.8	0.011108
68	1850	1845.1	0.26248
69	1850	1848.6	0.077395
70	1850	1844	0.32356
71	1850	1841	0.48475
72	1850	1846.3	0.19843
73	1850	1843.4	0.35772

Mu	allowance	Emission	% Percent Error
74	1850	1839	0.59252
75	1850	1845.7	0.23452
76	1850	1841.4	0.46631
77	1850	1847.9	0.11499
78	1850	1843.6	0.34384
79	1850	1840.9	0.49331
80	1850	1845.8	0.22495
81	1850	1843.1	0.37282
82	1850	1848	0.10948
83	1850	1845.3	0.25578
84	1850	1841.3	0.47222
85	1850	1844.3	0.3064
86	1850	1840.4	0.519
87	1850	1846.4	0.19328
88	1850	1842.5	0.40344
89	1850	1840	0.54026
90	1850	1844.6	0.29106
91	1850	1842.1	0.42658
92	1850	1846.6	0.18172
93	1850	1844.2	0.31595
94	1850	1840.5	0.51565
95	1850	1846.1	0.20827
96	1850	1842.5	0.40583
97	1850	1848.1	0.10341
98	1850	1844.5	0.29889
99	1850	1842.1	0.42617
100	1850	1846.4	0.1947
101	1850	1841.4	0.46656
102	1850	1837.9	0.65386
103	1850	1843.3	0.36191
104	1850	1839.9	0.54735
105	1850	1845.2	0.25988
106	1850	1841.8	0.4435
107	1850	1839.6	0.56258
108	1850	1843.7	0.34221
109	1850	1841.5	0.46034
110	1850	1845.5	0.24339
111	1850	1843.3	0.36058
112	1850	1840.1	0.53621
113	1850	1845.1	0.2632
114	1850	1841.9	0.43721
115	1850	1846.9	0.16813
116	1850	1843.7	0.34054
117	1850	1839.2	0.58327
118	1850	1843	0.37614
119	1850	1841	0.48639
120	1850	1837.9	0.65243
121	1850	1842.8	0.39172
122	1850	1839.7	0.55632
123	1850	1844.5	0.29921
124	1850	1841.4	0.46238
125	1850	1839.5	0.56793
126	1850	1843.1	0.37054
127	1850	1841.2	0.47536
128	1850	1844.8	0.28073
129	1850	1842.9	0.38483
130	1850	1840	0.5417
131	1850	1844.5	0.29626
132	1850	1841.6	0.45185
133	1850	1843.9	0.32833
134	1850	1841.1	0.48198
135	1850	1839.2	0.58123
136	1850	1842.7	0.39458
137	1850	1840.9	0.49319
138	1850	1838.1	0.64241
139	1850	1842.5	0.407
140	1850	1839.7	0.55506
141	1850	1844	0.32258
142	1850	1841.3	0.4695
143	1850	1839.6	0.56436
144	1850	1842.9	0.38568
145	1850	1841.1	0.47997
146	1850	1844.4	0.30356

Mu	allowance	Emission	% Percent Error
146	1850	1844.4	0.30356
147	1850	1842.7	0.39727
148	1850	1840	0.53908
149	1850	1842.1	0.42489
150	1850	1839.5	0.5651
151	1850	1843.6	0.3443
152	1850	1841.1	0.48349
153	1850	1839.4	0.57326
154	1850	1842.5	0.40346
155	1850	1840.9	0.49273
156	1850	1838.4	0.62828
157	1850	1842.3	0.41373
158	1850	1839.9	0.54833
159	1850	1843.8	0.33623
160	1850	1841.3	0.4699
161	1850	1839.7	0.55608
162	1850	1842.7	0.39293
163	1850	1841.1	0.47865
164	1850	1844.1	0.31739
165	1850	1840.7	0.50284
166	1850	1838.3	0.63179
167	1850	1842.1	0.42726
168	1850	1839.7	0.55536
169	1850	1843.5	0.35305
170	1850	1841.1	0.4803
171	1850	1839.6	0.56229
172	1850	1842.5	0.40659
173	1850	1841	0.48816
174	1850	1838.7	0.61237
175	1850	1842.3	0.41533
176	1850	1840	0.53876
177	1850	1843.6	0.34378
178	1850	1841.4	0.46642
179	1850	1839.9	0.54542
180	1850	1842.7	0.39533
181	1850	1839.5	0.56689
182	1850	1842.3	0.418
183	1850	1840.8	0.49581
184	1850	1838.6	0.61448
185	1850	1842.1	0.42593
186	1850	1839.9	0.54388
187	1850	1838.5	0.61971
188	1850	1841.2	0.47446
189	1850	1839.8	0.54993
190	1850	1842.5	0.40619
191	1850	1841.1	0.48131
192	1850	1839	0.59597
193	1850	1842.3	0.41381
194	1850	1840.2	0.52779
195	1850	1843.6	0.3474
196	1850	1841.5	0.46072
197	1850	1838.5	0.62034
198	1850	1841.1	0.48104
199	1850	1839.8	0.55329
200	1850	1842.3	0.41538

Minimal Emissions Economic Dispatch

I did not consider start-up costs per the formulation given in class lecture on exam.. This only considers the minimal incremental costs of the unit parameters and the minimization of emissions.

```
percenterror =

   0.0037604


totalemissions =

      1849.9


allowance =

       1850


mu =

   28


pedc =

           0        300          0
      387.18     211.76          0
      387.18     211.76        200
         300        100          0


u =

   0   1   0
   1   1   0
   1   1   1
   1   1   0


lamedc =

        70.3
```

```
  ECE 564   Spring 2007
  Exam # 1 -Midterm
  Due March 22, 2007
  Kathleen E. Williams
  Problem #4
 ---------------------------------------------------------------------------

echo off
clear

  fuel consumption parameters times the price (5.0 $/MBTU)
a=[10,20,30];
b=[6,8,9];
c=[0.005,0.0025,0.002];

  hourly load and reserve
pload=[300,600,800,400];

  units min, max values
pmin=[100,100,50];
pmax=[600,400,200];

  unit commitment is known
u = [0 1 0; 1 1 0;1 1 1 ;1 1 0];

  emissions in lbs/ MBTU
emissions = [200, 250, 120]/2000;
  emissions in lbs
allowance = 1850;

  determines lambda search range for economic dispatch in primal problem
for j=1:3
    lammin(j) = 5*b(j)+ 5*2*c(j)*pmin(j);
    lammax(j) = 5*b(j)+ 5*2*c(j)*pmax(j);
end

results = [];
  initializes the lamda and mu dual variables for the lagrange relaxation
mumax = 200;
mumin = 0;
     for t=1:4
       pcapmin(t)=0;
       lamsmin(t)=10000;
       pcapmax(t)=0;
       lamsmax(t)=-10000;
       for j=1:3
           if u(t,j) == 1
               if lamsmin(t) > lammin(j)
                   lamsmin(t) = lammin(j);
               end
               if lamsmax(t) < lammax(j)
                   lamsmax(t) = lammax(j);
               end
           end
       end
     end


for i=1:200
    mu = i;
    for t=1:4
           lamedc = (lamsmin(t) + lamsmax(t))/2;
           dellam = 0.1;
           for s=1:10000
               pedctot = 0;
               for j=1:3
                   if u(t,j) == 1
                       pedc(t,j) = (-5*b(j)-
mu*emissions(j)*b(j)+lamedc)/(10*c(j)+2*emissions(j)*mu*c(j));
                       if pedc(t,j) < pmin(j)
                           pedc(t,j) = pmin(j);
```

```
                    end
                    if pedc(t,j) > pmax(j)
                        pedc(t,j) = pmax(j);
                    end;
                else
                    pedc(t,j)=0;
                end;
                    pedctot = pedctot + pedc(t,j);
            end
              update the lambda
            if pedctot > pload(t)
                lamedc =  lamedc - dellam;
            end
            if pedctot < pload(t)
                lamedc = lamedc + dellam;
            end
        end

        sumemissions(t) = 0;
        for j=1:3
             sumemissions(t) = sumemissions(t) + emissions(j)*[a(j)+ b(j)*pedc(t,j)+
c(j)*pedc(t,j)^2]*u(t,j);
        end
    end
    totalemissions = 0;
    for t=1:4
    totalemissions = totalemissions + sumemissions(t);
    end
    percenterror = 100*(allowance-totalemissions)/allowance;
if abs(allowance-totalemissions) < 1
    percenterror = 100*(allowance-totalemissions)/allowance
    totalemissions
    mu
else
end
     results = [results; mu allowance totalemissions percenterror];
end
```

Problem # 5

Given the optimal unit commitment solution in Table 4, solve the economic dispatch for the peak load hour considering the power flow constraint.

Formulation

Line 13 $F_{13} = \frac{2}{3} P_1 + \frac{1}{3} P_2 \leq 310$ MW

$$\mathcal{L} = \min \sum_{t=1}^{4} \sum_{i=1}^{3} F_i(P_i) u_{i,t} + \mu\left[\frac{2}{3} P_{1,t} + \frac{1}{3} P_{2,t} - 310\right]$$

$$+ \sum_{t=1}^{4} \lambda^t \left(P_{load} - \sum_{i=1}^{3} P_i^t u_i\right)$$

unit commitment

0 1 0

1 1 0

1 1 1

0

$P_i^{min} \leq P_i \leq P_i^{max}$

$$P_{1,t} = \frac{\lambda - \frac{2}{3}\mu - b_1}{2c_1}$$

$$P_{2,t} = \frac{\lambda - \frac{1}{3}\mu - b_2}{2c_2}$$

$$P_{3,t} = \frac{\lambda - b_3}{2c_3}$$

Results

u =

0	1	0
1	1	0
1	1	1
1	1	0

pedc = Economic Dispatch

0	300.04	0
320.02	280.04	0
329.1	298.2	172.75
253.34	146.68	0

mu =

3 μ

lamedc =

44.666 λ

flow = Line 1-3 MW Flows $f_{13} = \frac{2}{3} P_1 + \frac{1}{3} P_2$

100.01	306.69	318.8	217.79

```
-------------------------------------------------------------------
 ECE 564   Spring 2007
 Exam # 1 Midterm
 Due March 12, 2007
 Kathleen B. Williams
 Problem #5
-------------------------------------------------------------------

echo off
clear

 fuel consumption parameters times the price 5.5 $/MBTU
a=[10,20,30]*5;
b=[6,8,9]*5;
c=[0.005,0.0025,0.002]*5;

 hourly load and reserve
pload=[300,600,800,400];

 units min, max values
pmin=[100,100,50];
pmax=[600,400,200];

 unit commitment is known
u = [0 1 0; 1 1 0;1 1 1 ;1 1 0];

 determines lambda search range for economic dispatch in primal problem
for j=1:3
    lammin(j) = b(j)+ 2*c(j)*pmin(j);
    lammax(j) = b(j)+ 2*c(j)*pmax(j);
end

limit = 320;
results = [];
 initialises the lamda and mu dual variables for the Lagrange relaxation
     for t=1:4
       pcapmin(t)=0;
       lamsmin(t)=10000;
       pcapmax(t)=0;
       lamsmax(t)=-10000;
       for j=1:3
           if u(t,j) == 1
               if lamsmin(t) > lammin(j)
                   lamsmin(t) = lammin(j);
               end
               if lamsmax(t) < lammax(j)
                   lamsmax(t) = lammax(j);
               end
           end
       end
     end

    mu=1;
    for i=1:100
    for t=1:4
           lamedc = (lamsmin(t) + lamsmax(t))/2;
           dellam = 0.001;
           for j=1:3
              pedc(t,j) = 0;
           end
           for s=1:10000
               pedctot = 0;
               for j=1:3
                   if u(t,j) == 1 && j == 1
                       pedc(t,j) = (lamedc -(2/3)*mu - b(j))/(2*c(j));
                       if pedc(t,j) < pmin(j)
                           pedc(t,j) = pmin(j);
                       end
                       if pedc(t,j) > pmax(j)
```

```
                    pedc(t,j) = pmax(j);
                end;
            else
            end;
            if u(t,j) == 1 && j == 2
                pedc(t,j) = (lamedc -(1/3)*mu - b(j))/(2*c(j));
                if pedc(t,j) < pmin(j)
                    pedc(t,j) = pmin(j);
                end
                if pedc(t,j) > pmax(j)
                    pedc(t,j) = pmax(j);
                end;
            else
            end;
            if u(t,j) == 1 && j == 3
                pedc(t,j) = (lamedc - b(j))/(2*c(j));
                if pedc(t,j) < pmin(j)
                    pedc(t,j) = pmin(j);
                end
                if pedc(t,j) > pmax(j)
                    pedc(t,j) = pmax(j);
                end;
            else
            end;

            pedctot = pedctot + pedc(t,j);
        end
          update the lambda
        if pedctot > pload(t)
            lamedc =  lamedc - dellam;
        end
        if pedctot < pload(t)
            lamedc = lamedc + dellam;
        end
        flow(t) = ((2/3)*pedc(t,1) + (1/3)*pedc(t,2));
        if flow(t) > limit
           mu = mu+1;
        else
        end

    end
end
end
u
pedc
mu
lamedc
flow
```