ภาคผนวก ค.

โปรแกรมคอมพิวเตอร์

ค.1 วิธีไดแนมิคโปรแกรมมิ่งและเทคนิคการปรับซูม

```
      program ceddynz
*
*     This program improves dynamic programming algorithms
*     for constrained economic generation dispatch problems.
*
      implicit none

      integer maxgen, maxlen, szdcmw, szdcdm, maxiter, iter, stage
      integer indexv, i, j, k, pos, lindex, hrfn, m, n
      real rampup, rampdn, delta, kdelta, mwnow, lowlim, hilim, pf
      real dcmw, dcdm, dcmwtmp, dcdmtmp, hyyxx, pyyxx, pxxxx
      real edemand, load, dltasz, odltasz, bcoeff, b0, b00
      real mwtmp1, mwtmp2, sumlow, sumhi, heattmp, losstmp
      real sumheat, power, sysloss, heatrate, mwlow, mwhi
      real summw, sumcost, odemand, osysloss, osummw, ltime
      real etime, elapsed(2), total

      parameter (maxgen = 70)
      parameter (delta = 2.0, kdelta = 2.0)
      parameter (maxlen = (240)/delta)

      dimension mwnow(1:maxgen), lowlim(1:maxgen), hilim(1:maxgen)
      dimension mwlow(1:maxgen), mwhi(1:maxgen)
      dimension rampdn(1:maxgen), rampup(1:maxgen)
      dimension pf(1:maxgen), power(1:maxgen), szdcdm(1:maxgen-1)
      dimension dcmw(maxgen, maxlen+1), szdcmw(1:maxgen)
      dimension dcmwtmp(1:maxlen+1), dcdmtmp(1:maxgen*maxlen+1)
      dimension dcdm(maxgen-1, maxgen*maxlen+1)
      dimension hyyxx(maxgen-1, maxgen*maxlen+1)
      dimension pyyxx(maxgen-1, maxgen*maxlen+1)
      dimension pxxxx(maxgen-1, maxgen*maxlen+1)
      dimension bcoeff(maxgen, maxgen), b0(1:maxgen), losstmp(1:maxgen)
      dimension hrfn(1:maxgen)

*      data mwnow/maxgen*400.0/

*      data mwnow/maxgen*400.0/
*      data rampup/maxgen*50.0/
*      data rampdn/maxgen*80.0/

       mwnow(1) = 70.0
       mwnow(2) = 79.0
       mwnow(3) = 79.0
       mwnow(4) = 79.0
       mwnow(5) = 150.0
       mwnow(6) = 150.0
       mwnow(7) = 90.0
       mwnow(8) = 90.0
       mwnow(9) = 90.0
       mwnow(10) = 90.0
       mwnow(11) = 112.0
       mwnow(12) = 111.0
       mwnow(13) = 102.0
       mwnow(14) = 152.0
       mwnow(15) = 115.0
       mwnow(16) = 76.0
       mwnow(17) = 76.0
       mwnow(18) = 76.0
       mwnow(19) = 180.0
       mwnow(20) = 700.0
       mwnow(21) = 320.0
       mwnow(22) = 693.0
       mwnow(23) = 284.0
       mwnow(24) = 140.0
       mwnow(25) = 140.0
```

```
mwnow(26) = 140.0
mwnow(27) = 88.0
mwnow(28) = 88.0
mwnow(29) = 88.0
mwnow(30) = 120.0
mwnow(31) = 120.0
mwnow(32) = 120.0
mwnow(33) = 180.0
mwnow(34) = 20.0
mwnow(35) = 20.0
mwnow(36) = 700.0
mwnow(37) = 700.0
mwnow(38) = 105.0
mwnow(39) = 105.0
mwnow(40) = 90.0
mwnow(41) = 63.0
mwnow(42) = 63.0
mwnow(43) = 90.0
mwnow(44) = 25.0
mwnow(45) = 10.0
mwnow(46) = 32.0
mwnow(47) = 55.0
mwnow(48) = 55.0
mwnow(49) = 90.0
mwnow(50) = 60.0
mwnow(51) = 60.0
mwnow(52) = 90.0
mwnow(53) = 90.0
mwnow(54) = 90.0
mwnow(55) = 90.0
mwnow(56) = 90.0
mwnow(57) = 90.0
mwnow(58) = 180.0
mwnow(59) = 250.0
mwnow(60) = 70.0
mwnow(61) = 292.0
mwnow(62) = 292.0
mwnow(63) = 76.0
mwnow(64) = 76.0
mwnow(65) = 76.0
mwnow(66) = 673.0
mwnow(67) = 673.0
mwnow(68) = 350.0
mwnow(69) = 350.0
mwnow(70) = 350.0

rampup(1) = 5.0*6.8
rampup(2) = 5.0*40.0
rampup(3) = 5.0*40.0
rampup(4) = 5.0*40.0
rampup(5) = 5.0*5.0
rampup(6) = 5.0*5.0
rampup(7) = 5.0*3.0
rampup(8) = 5.0*3.0
rampup(9) = 5.0*3.0
rampup(10) = 5.0*3.0
rampup(11) = 5.0*70.0
rampup(12) = 5.0*70.0
rampup(13) = 5.0*70.0
rampup(14) = 5.0*120.0
rampup(15) = 5.0*115.0
rampup(16) = 5.0*76.3
rampup(17) = 5.0*76.3
rampup(18) = 5.0*76.3
rampup(19) = 5.0*8.0
rampup(20) = 5.0*39.0
rampup(21) = 5.0*22.5
rampup(22) = 5.0*25.0
rampup(23) = 5.0*10.0
rampup(24) = 5.0*35.0
rampup(25) = 5.0*35.0
rampup(26) = 5.0*35.0
rampup(27) = 5.0*87.0
rampup(28) = 5.0*87.0
rampup(29) = 5.0*87.0
rampup(30) = 5.0*70.2
rampup(31) = 5.0*70.2
rampup(32) = 5.0*70.2
rampup(33) = 5.0*70.2
rampup(34) = 5.0*20.0
```

```
rampup(35) = 5.0*20.0
rampup(36) = 5.0*39.0
rampup(37) = 5.0*39.0
rampup(38) = 5.0*70.0
rampup(39) = 5.0*70.0
rampup(40) = 5.0*35.0
rampup(41) = 5.0*63.0
rampup(42) = 5.0*63.0
rampup(43) = 5.0*35.0
rampup(44) = 5.0*35.0
rampup(45) = 5.0*10.0
rampup(46) = 5.0*35.0
rampup(47) = 5.0*35.0
rampup(48) = 5.0*35.0
rampup(49) = 5.0*35.0
rampup(50) = 5.0*35.0
rampup(51) = 5.0*35.0
rampup(52) = 5.0*35.0
rampup(53) = 5.0*35.0
rampup(54) = 5.0*35.0
rampup(55) = 5.0*35.0
rampup(56) = 5.0*35.0
rampup(57) = 5.0*35.0
rampup(58) = 5.0*8.0
rampup(59) = 5.0*39.0
rampup(60) = 5.0*6.8
rampup(61) = 5.0*55.0
rampup(62) = 5.0*55.0
rampup(63) = 5.0*76.3
rampup(64) = 5.0*76.3
rampup(65) = 5.0*76.3
rampup(66) = 5.0*10.0
rampup(67) = 5.0*10.0
rampup(68) = 5.0*8.0
rampup(69) = 5.0*8.0
rampup(70) = 5.0*8.0

rampdn(1) = 5.0*6.8
rampdn(2) = 5.0*40.0
rampdn(3) = 5.0*40.0
rampdn(4) = 5.0*40.0
rampdn(5) = 5.0*5.0
rampdn(6) = 5.0*5.0
rampdn(7) = 5.0*3.0
rampdn(8) = 5.0*3.0
rampdn(9) = 5.0*3.0
rampdn(10) = 5.0*3.0
rampdn(11) = 5.0*70.0
rampdn(12) = 5.0*70.0
rampdn(13) = 5.0*70.0
rampdn(14) = 5.0*120.0
rampdn(15) = 5.0*115.0
rampdn(16) = 5.0*76.3
rampdn(17) = 5.0*76.3
rampdn(18) = 5.0*76.3
rampdn(19) = 5.0*8.0
rampdn(20) = 5.0*39.0
rampdn(21) = 5.0*22.5
rampdn(22) = 5.0*25.0
rampdn(23) = 5.0*10.0
rampdn(24) = 5.0*35.0
rampdn(25) = 5.0*35.0
rampdn(26) = 5.0*35.0
rampdn(27) = 5.0*87.0
rampdn(28) = 5.0*87.0
rampdn(29) = 5.0*87.0
rampdn(30) = 5.0*70.2
rampdn(31) = 5.0*70.2
rampdn(32) = 5.0*70.2
rampdn(33) = 5.0*70.2
rampdn(34) = 5.0*20.0
rampdn(35) = 5.0*20.0
rampdn(36) = 5.0*39.0
rampdn(37) = 5.0*39.0
rampdn(38) = 5.0*70.0
rampdn(39) = 5.0*70.0
rampdn(40) = 5.0*35.0
rampdn(41) = 5.0*63.0
rampdn(42) = 5.0*63.0
rampdn(43) = 5.0*35.0
```

```
        rampdn(44) = 5.0*35.0
        rampdn(45) = 5.0*10.0
        rampdn(46) = 5.0*35.0
        rampdn(47) = 5.0*35.0
        rampdn(48) = 5.0*35.0
        rampdn(49) = 5.0*35.0
        rampdn(50) = 5.0*35.0
        rampdn(51) = 5.0*35.0
        rampdn(52) = 5.0*35.0
        rampdn(53) = 5.0*35.0
        rampdn(54) = 5.0*35.0
        rampdn(55) = 5.0*35.0
        rampdn(56) = 5.0*35.0
        rampdn(57) = 5.0*35.0
        rampdn(58) = 5.0*8.0
        rampdn(59) = 5.0*39.0
        rampdn(60) = 5.0*6.8
        rampdn(61) = 5.0*55.0
        rampdn(62) = 5.0*55.0
        rampdn(63) = 5.0*76.3
        rampdn(64) = 5.0*76.3
        rampdn(65) = 5.0*76.3
        rampdn(66) = 5.0*10.0
        rampdn(67) = 5.0*10.0
        rampdn(68) = 5.0*8.0
        rampdn(69) = 5.0*8.0
        rampdn(70) = 5.0*8.0

*       loadf(1) = 0.995
*       loadf(2) = 0.979
*       loadf(3) = 0.970
*       loadf(4) = 0.985
*       loadf(5) = 0.992
*       loadf(6) = 1.015

      data pf/maxgen*1.0/

      open(unit = 1, file = 'b.dat', status = 'old')
      do 1 i = 1, maxgen
        do 2 j = 1, maxgen
           read (1, *) bcoeff(i, j)
    2    continue
    1 continue
      close(1)

      open(unit = 2, file = 'b0.dat', status = 'old')
      do 3 i = 1, maxgen
        read (2, *) b0(i)
    3 continue
      read(2,*) b00
      close(2)

      do 4 i = 1, maxgen
        hrfn(i) = i
    4 continue

      lowlim(1) = 60.0
      lowlim(2) = 50.0
      lowlim(3) = 50.0
      lowlim(4) = 50.0
      lowlim(5) = 150.0
      lowlim(6) = 150.0
      lowlim(7) = 90.0
      lowlim(8) = 90.0
      lowlim(9) = 90.0
      lowlim(10) = 90.0
      lowlim(11) = 60.0
      lowlim(12) = 60.0
      lowlim(13) = 60.0
      lowlim(14) = 60.0
      lowlim(15) = 60.0
      lowlim(16) = 40.0
      lowlim(17) = 40.0
      lowlim(18) = 40.0
      lowlim(19) = 180.0
      lowlim(20) = 350.0
      lowlim(21) = 315.0
      lowlim(22) = 280.0
      lowlim(23) = 284.0
      lowlim(24) = 140.0
```

```
lowlim(25) = 140.0
lowlim(26) = 140.0
lowlim(27) = 50.0
lowlim(28) = 50.0
lowlim(29) = 50.0
lowlim(30) = 70.0
lowlim(31) = 70.0
lowlim(32) = 70.0
lowlim(33) = 120.0
lowlim(34) = 15.0
lowlim(35) = 15.0
lowlim(36) = 250.0
lowlim(37) = 250.0
lowlim(38) = 20.0
lowlim(39) = 20.0
lowlim(40) = 59.0
lowlim(41) = 10.0
lowlim(42) = 10.0
lowlim(43) = 59.0
lowlim(44) = 17.0
lowlim(45) = 5.0
lowlim(46) = 21.0
lowlim(47) = 36.0
lowlim(48) = 36.0
lowlim(49) = 59.0
lowlim(50) = 39.0
lowlim(51) = 39.0
lowlim(52) = 59.0
lowlim(53) = 59.0
lowlim(54) = 59.0
lowlim(55) = 59.0
lowlim(56) = 59.0
lowlim(57) = 59.0
lowlim(58) = 180.0
lowlim(59) = 250.0
lowlim(60) = 60.0
lowlim(61) = 250.0
lowlim(62) = 250.0
lowlim(63) = 40.0
lowlim(64) = 40.0
lowlim(65) = 40.0
lowlim(66) = 200.0
lowlim(67) = 200.0
lowlim(68) = 100.0
lowlim(69) = 100.0
lowlim(70) = 100.0

hilim(1) = 70.0
hilim(2) = 79.0
hilim(3) = 79.0
hilim(4) = 79.0
hilim(5) = 276.0
hilim(6) = 276.0
hilim(7) = 140.0
hilim(8) = 140.0
hilim(9) = 140.0
hilim(10) = 140.0
hilim(11) = 112.0
hilim(12) = 111.0
hilim(13) = 102.0
hilim(14) = 152.0
hilim(15) = 115.0
hilim(16) = 76.0
hilim(17) = 76.0
hilim(18) = 76.0
hilim(19) = 330.0
hilim(20) = 700.0
hilim(21) = 360.0
hilim(22) = 700.0
hilim(23) = 316.0
hilim(24) = 735.0
hilim(25) = 735.0
hilim(26) = 735.0
hilim(27) = 88.0
hilim(28) = 88.0
hilim(29) = 88.0
hilim(30) = 120.0
hilim(31) = 120.0
hilim(32) = 120.0
hilim(33) = 180.0
```

```
      hilim(34) = 20.0
      hilim(35) = 20.0
      hilim(36) = 700.0
      hilim(37) = 700.0
      hilim(38) = 105.0
      hilim(39) = 105.0
      hilim(40) = 90.0
      hilim(41) = 63.0
      hilim(42) = 63.0
      hilim(43) = 90.0
      hilim(44) = 25.0
      hilim(45) = 10.0
      hilim(46) = 32.0
      hilim(47) = 55.0
      hilim(48) = 55.0
      hilim(49) = 90.0
      hilim(50) = 60.0
      hilim(51) = 60.0
      hilim(52) = 90.0
      hilim(53) = 90.0
      hilim(54) = 90.0
      hilim(55) = 90.0
      hilim(56) = 90.0
      hilim(57) = 90.0
      hilim(58) = 330.0
      hilim(59) = 700.0
      hilim(60) = 70.0
      hilim(61) = 292.0
      hilim(62) = 292.0
      hilim(63) = 76.0
      hilim(64) = 76.0
      hilim(65) = 76.0
      hilim(66) = 673.0
      hilim(67) = 673.0
      hilim(68) = 350.0
      hilim(69) = 350.0
      hilim(70) = 350.0

*
*     Determining the system loss.
*         System loss = PiBijPj + Bi0Pi + B00
*
            sysloss = 0.0
            summw = 0.0
            sumcost = 0.0
            do 6140 m = 1, maxgen
               losstmp(m) = 0.0
               summw = summw + mwnow(m)
               sumcost = sumcost + heatrate(hrfn(m),
     + mwnow(m), lowlim(m), hilim(m))
               do 6145 n = 1, maxgen
                  losstmp(m) = losstmp(m) +
     +                    mwnow(n)*bcoeff(n,m)/100.0
 6145          continue
               sysloss = sysloss + losstmp(m)*mwnow(m)/100.0
 6140       continue

            do 6150 m = 1, maxgen
               sysloss = sysloss + mwnow(m)*b0(m)/100.0
 6150       continue
            sysloss = (sysloss + b00)*100.0
            odemand = summw - sysloss
      osysloss = sysloss
      osummw = summw
      maxiter = 10
      edemand = 1.008*odemand
      load = edemand
      dltasz = delta
      odltasz = dltasz
      print *, 'summw = ', summw,
     +   ' odemand = ',  odemand,
     +   ' %loss = ', sysloss/summw*100.0
*      pause

      total = etime(elapsed)
      ltime = elapsed(1)
*
*     Discretizing unit megawatts
*
      do 10 i = 1, maxgen
```

```
         mwtmp1 = mwnow(i) - rampdn(i)
         mwtmp2 = mwnow(i) + rampup(i)
*        print *,'u', i, lowlim(i), mwnow(i), hilim(i)
         call chkmw(mwtmp1, lowlim(i), hilim(i))
         call chkmw(mwtmp2, lowlim(i), hilim(i))
         mwlow(i) = mwtmp1
         mwhi(i) = mwtmp2
*        print *,'u', i, mwlow(i), mwnow(i), mwhi(i)
         call discre(dcmwtmp, maxlen+1, mwtmp1, mwtmp2, dltasz,
     +               szdcmw, maxgen, i)
         do 5 j = 1, szdcmw(i)
            dcmw(i, j) = dcmwtmp(j)
    5    continue
*        print *,'u', i, dcmw(i, 1), mwnow(i), dcmw(i, szdcmw(i))
   10 continue
*
*     Discretizing demands
*
      do 20 i = 1, maxgen-1
         if (i .eq. 1) then
            sumlow = dcmw(1, 1) + dcmw(2, 1)
            sumhi = dcmw(1, szdcmw(1)) + dcmw(2, szdcmw(2))
         else
            sumlow = sumlow + dcmw(i+1, 1)
            sumhi = sumhi + dcmw(i+1, szdcmw(i+1))
         end if
         call discre(dcdmtmp, maxgen*maxlen+1, sumlow, sumhi, dltasz,
     +               szdcdm, maxgen-1, i)
         do 15 j = 1, szdcdm(i)
            dcdm(i, j) = dcdmtmp(j)
   15    continue

*        print *, 'Load', dcdm(i, 1),  dcdm(i, szdcdm(i))
   20 continue

*     ************************************************************
*                           Stage calculation
*     ************************************************************

      do 900 iter = 1, maxiter
         if (dltasz .lt. 0.01) goto 905
*         print *, '********** Epoch : ', iter, ' **********'
*         print *, 'Delta = ', dltasz

         do 910 i = 1, maxgen-1
            do 915 j = 1, maxgen*maxlen+1
               hyyxx(i, j) = 1000000000.0
               pyyxx(i, j) = 0.0
               pxxxx(i, j) = 0.0
  915       continue
  910    continue

*         print *, 'Performing stage 1 ...'
         stage = 1
         do 920  i = 1, szdcdm(stage)
            do 925 j = 1, szdcmw(stage+1)
*              print *, dcdm(stage, i), dcmw(stage+1, j),
*     + (dcdm(stage, i) - dcmw(stage+1, j)), dcmw(stage,1)
               if ((dcdm(stage, i)-dcmw(stage+1, j)) .lt. dcmw(stage,1))
     + goto 925
               heattmp = heatrate(hrfn(stage+1), dcmw(stage+1, j),
     + dcmw(stage+1, 1), dcmw(stage+1, szdcmw(stage+1)))*
     + abs(pf(stage+1)) +
     + heatrate(hrfn(stage), dcdm(stage, i)-dcmw(stage+1, j),
     + dcmw(stage, 1), dcmw(stage, szdcmw(stage)))*abs(pf(stage))
*        print *, '--------------------'
*        print *, hrfn(stage+1), dcmw(stage+1, j), dcmw(stage+1, 1),
*     + dcmw(stage+1, szdcmw(stage+1))
*        print *, heatrate(hrfn(stage+1), dcmw(stage+1, j),
*     + dcmw(stage+1, 1), dcmw(stage+1, szdcmw(stage+1)))*pf(stage+1)
*        print *, hrfn(stage), dcdm(stage, i)-dcmw(stage+1, j),
*     + dcmw(stage, 1), dcmw(stage, szdcmw(stage))
*        print *, heatrate(hrfn(stage), dcdm(stage, i)-dcmw(stage+1, j),
*     + dcmw(stage, 1), dcmw(stage, szdcmw(stage)))*pf(stage)
*        print *, 'Ht', heattmp
               if (heattmp .lt. hyyxx(stage, i)) then
                  hyyxx(stage, i) = heattmp
                  pyyxx(stage, i) = dcmw(stage+1, j)
                  pxxxx(stage, i) = dcdm(stage, i) - dcmw(stage+1, j)
*       print *, 's', stage, i, 1, pxxxx(stage, i), pyyxx(stage, i),
```

```
*     + heattmp
                end if
  925       continue
  920    continue

         do 935 stage = 2, maxgen-1
*           print *, 'Performing stage ', stage, ' ...'
            do 940  i = 1, szdcdm(stage)
               if (stage .eq. maxgen-1) then
                  if (i .le. (szdcdm(stage)-1)) then
*                           print *, dcdm(stage, i), load,
*     +                  dcdm(stage, i+1)
                     if ((load .ge. dcdm(stage, i)) .and. (load .lt.
     +                    dcdm(stage, i+1))) then
                        lindex = i
                     end if
                  end if
               end if

               if ((dcdm(stage, i)-dcmw(stage+1,1)) .le. dcdm(stage-1,
     +              szdcdm(stage-1))) then
                   pos = 1
               else
                   do 944 k = 1, maxlen+1
                      dcmwtmp(k) = dcmw(stage+1, k)
  944              continue
                   pos = indexv(dcmwtmp, maxlen+1, szdcmw(stage+1),
     +           (dcdm(stage, i) - dcdm(stage-1, szdcdm(stage-1))))
               end if

               do 945 j = pos, szdcmw(stage+1)
                  if ((dcdm(stage, i)-dcmw(stage+1, j)) .lt.
     +                dcdm(stage-1, 1)) goto 940
                  heattmp = heatrate(hrfn(stage+1), dcmw(stage+1, j),
     + dcmw(stage+1, 1), dcmw(stage+1, szdcmw(stage+1)))*
     + abs(pf(stage+1))
                  do 955 k = 1, maxgen*maxlen+1
                     dcdmtmp(k) = dcdm(stage-1, k)
  955             continue
                  pos = indexv(dcdmtmp, maxgen*maxlen+1,
     + szdcdm(stage-1), (dcdm(stage, i) - dcmw(stage+1, j)))
                  heattmp = heattmp + hyyxx(stage-1, pos)
*       print *, 's', stage, i, pos, hyyxx(stage-1, i), heattmp,
*     +  dcmw(stage+1, j), dcdm(stage, i) - dcmw(stage+1, j)
                  if (heattmp .lt. hyyxx(stage, i)) then
                     hyyxx(stage, i) = heattmp
                     pyyxx(stage, i) = dcmw(stage+1, j)
                     pxxxx(stage, i) = dcdm(stage, i) - dcmw(stage+1, j)
*       print *, 's', stage, i, pos, pxxxx(stage, i), pyyxx(stage, i),
*     + heattmp
                  end if
  945          continue
  940       continue
*          print *, 'Stage ', stage, ' completed ...'
  935    continue

*          print *, 'All stages completed...'
*
*     Interprets unit powers.
*
         stage = maxgen - 1
         power(maxgen) = pyyxx(stage, lindex)
         do 960 stage = maxgen-2, 1, -1
            do 965 i = 1, szdcdm(stage)
*                if ((pxxxx(stage+1, lindex) .ge. dcdm(stage, i)) .and.
*     +              (pxxxx(stage+1, lindex) .lt. dcdm(stage, i+1))) then
*        print *, 'stage', stage, lindex, pxxxx(stage+1, lindex), '=',
*     +  dcdm(stage, i), dcdm(stage, szdcdm(stage))
                  if (pxxxx(stage+1, lindex) .eq. dcdm(stage, i)) then
                     lindex = i
*        print *, 'We got it !', 'u', stage+1, i, pyyxx(stage, lindex),
*     +        mwlow(stage+1), mwhi(stage+1)
                     power(stage+1) = pyyxx(stage, lindex)
                     goto 960
                  end if
  965       continue
  960    continue
         power(1) = pxxxx(1, lindex)
*        print *, 'lindex = ', lindex, pxxxx(1, lindex)
*        print *, power
```

```
*       pause

*
*     Determine the system loss.
*         System loss = PiBijPj + Bi0Pi + B00
*
      sysloss = 0.0
      do 975 i = 1, maxgen
         losstmp(i) = 0.0
         do 980 j = 1, maxgen
            losstmp(i) = losstmp(i) + power(j)*bcoeff(j,i)/100.0
  980    continue
         sysloss = sysloss + losstmp(i)*power(i)/100.0
  975 continue

      do 985 i = 1, maxgen
         sysloss = sysloss + power(i)*b0(i)/100.0
  985 continue
      sysloss = (sysloss + b00)*100.0
*
*     Determine penalty factors.
*        Penalty = 1/(1-@L/@P)
*                = 1/(1-2*BijPj-Bi0)
*     where L  = PiBijPj + Bi0Pi + B00
*
      do 990 i = 1, maxgen
         pf(i) = 0.0
         losstmp(i) = 0.0
         do 995 j = 1, maxgen
            losstmp(i) = losstmp(i) + bcoeff(i, j)*power(j)/100.0
  995    continue
         pf(i) = 1.0/(1.0 - (2.0*losstmp(i) + b0(i)))
  990 continue
*      print *, 'PF = ', pf

      load = edemand + sysloss

      if (iter .ge. 2) then
         dltasz = dltasz / kdelta
*
*     Zooming...
*     Re-discretizes unit megawatts
*
      do 1000 i = 1, maxgen
         mwtmp1 = power(i) - dltasz*(dcmw(i, szdcmw(i)) - dcmw(i, 1))
     +            / (2.0*odltasz)
         mwtmp2 = power(i) + dltasz*(dcmw(i, szdcmw(i)) - dcmw(i, 1))
     +            / (2.0*odltasz)
         call chkmw(mwtmp1, mwlow(i), mwhi(i))
         call chkmw(mwtmp2, mwlow(i), mwhi(i))
         call discre(dcmwtmp, maxlen+1, mwtmp1, mwtmp2, dltasz,
     +              szdcmw, maxgen, i)
         do 1005 j = 1, szdcmw(i)
            dcmw(i, j) = dcmwtmp(j)
 1005    continue
*        print *, 'u', i, dcmw(i, 1), mwnow(i), power(i),
*     +          dcmw(i, szdcmw(i))
 1000 continue
*
*     Re-discretizes demands
*
      do 1010 i = 1, maxgen-1
         if (i .eq. 1) then
            sumlow = dcmw(1, 1) + dcmw(2, 1)
            sumhi = dcmw(1, szdcmw(1)) + dcmw(2, szdcmw(2))
         else
            sumlow = sumlow + dcmw(i+1, 1)
            sumhi = sumhi + dcmw(i+1, szdcmw(i+1))
         end if
         call discre(dcdmtmp, maxgen*maxlen+1, sumlow, sumhi, dltasz,
     +              szdcdm, maxgen-1, i)
         do 1015 j = 1, szdcdm(i)
            dcdm(i, j) = dcdmtmp(j)
 1015    continue
 1010 continue

      odltasz = dltasz

      end if
*
```

```
*     end of zooming phase.
*
  900 continue
      total = etime(elapsed)
  905 print *, 'Calculation completed !...'
*      print *, power

*      do 919 i = 1, maxgen
*        print *, 'Unit ', i, ' : ', heatrate(hrfn(i), power(i),
*     + mwlow(i), mwhi(i))
*   919 continue

      open(unit = 3, file = 'output.dat', status = 'new')
      write (3, *) 'Total time = ', elapsed(1)-ltime
      mwtmp1 = 0.0
      sumheat = 0.0
        write (3, *) 'P  mw(i) cost mw(i) cost'
      do 6015 i = 1, maxgen
        mwtmp1 = mwtmp1 + power(i)
        sumheat = sumheat +  heatrate(hrfn(i), power(i), lowlim(i),

     + hilim(i))
        write (3, 512) i, mwnow(i),
     +      heatrate(hrfn(i), mwnow(i), lowlim(i), hilim(i)),
     +      power(i),
     +      heatrate(hrfn(i), power(i), lowlim(i), + hilim(i))
  512   format (i3, f15.5, f15.5, f15.5, f15.5)
 6015 continue
      write (3, *) 'sum mw (old) = ', summw,
     +  ' ->  new = ', mwtmp1
      write (3, *) 'Sum cost (old) = ', sumcost,
     +  ' ->  new = ', sumheat
      write (3, *) 'System loss = ', osysloss, ' (',
     + osysloss/osummw*100.0, '%) ->  new = ', sysloss, ' (',
     + sysloss/mwtmp1*100.0, '%)'
      write (3, *) 'diff = ', mwtmp1 - sysloss - edemand
      write (3, *) 'Demand (old) = ', odemand,
     +  ' -> new = ', edemand
      write (3, *) 'delta = ', delta
      close(3)

      stop
      end

      subroutine chkmw(megawt, mwlow, mwhi)
      implicit none
      real megawt, mwlow, mwhi
      if (megawt .lt. mwlow) then
        megawt = mwlow
      else if (megawt .gt. mwhi) then
        megawt = mwhi
      end if
      return
      end

      subroutine discre(dcval, szdcval, minv, maxv, dltasz,
     +                  aszdc, szaszdc, no)
      implicit none
      integer szdcval, szaszdc, no, aszdc(1:szaszdc), i
      real dcval(1:szdcval), minv, maxv, dltasz, temp

      do 9000 i = 1, szdcval
        temp = minv + dltasz*(i-1)
        if (temp .eq. maxv) then
           dcval(i) = temp
           aszdc(no) = i
           goto 9005
        else if (temp .gt. maxv) then
           aszdc(no) = i - 1
           goto 9005
        else
           dcval(i) = temp
           aszdc(no) = i
        end if
 9000 continue
 9005 return
      end

      function heatrate(no, megawt, mwlow, mwhi)
```

```
      implicit none
      integer no
      real heatrate, megawt, mwlow, mwhi, gcost, ocost
      real trfactor, opfactor
      real coef_a, coef_b, coef_c

*     Gas & Oil (Baht/GCal)
      gcost = 500.25
      ocost = 602.73

*     Transform BTU->Cal
      trfactor = 251.996

      if (megawt .lt. mwlow) then
        heatrate = 1000000000.0
        goto 9010
      else if (megawt .gt. mwhi) then
        heatrate = 1000000000.0
        goto 9010
      end if

      if (no .eq. 1) then
*     KN-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.010
      goto 9110
      else if (no .eq. 2) then
*     RPB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.015
      goto 9110
      else if (no .eq. 3) then
*     RPB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 4) then
*     RPB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 5) then
*     MM-T8
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.010
      goto 9110
      else if (no .eq. 6) then
*     MM-T9
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.015
      goto 9110
      else if (no .eq. 7) then
*     MM-T4
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.000
      goto 9110
      else if (no .eq. 8) then
*     MM-T5
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.010
      goto 9110
      else if (no .eq. 9) then
*     MM-T6
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.015
      goto 9110
      else if (no .eq. 10) then
*     MM-T7
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.020
```

```
      goto 9110
      else if (no .eq. 11) then
*     SK-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.030
      goto 9110
      else if (no .eq. 12) then
*     SK-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.035
      goto 9110
      else if (no .eq. 13) then
*     SK-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.040
      goto 9110
      else if (no .eq. 14) then
*     BB-H8
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.045
      goto 9110
      else if (no .eq. 15) then
*     BB-H7
      coef_a = 0.0

      coef_b = 0.01
      coef_c = 0.050
      goto 9110
      else if (no .eq. 16) then
*     BB-H6
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.055
      goto 9110
      else if (no .eq. 17) then
*     BB-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.060
      goto 9110
      else if (no .eq. 18) then
*     BB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.065
      goto 9110
      else if (no .eq. 19) then
*     NPO-C1
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
      else if (no .eq. 20) then
*     IPT-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.070
      goto 9110
      else if (no .eq. 21) then
*     BPK-C1
      coef_a = 22.4
      coef_b = 5760.0
      coef_c = 222000.0
      goto 9110
      else if (no .eq. 22) then
*     BPK-T2
      coef_a = 1.56
      coef_b = 8734.0
      coef_c = 59400.0
      goto 9110
      else if (no .eq. 23) then
*     SB-C1
      coef_a = -0.68
      coef_b = 6653.0
      coef_c = 114037.0
      goto 9110
```

```
      else if (no .eq. 24) then
*     SB-T1
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.0
      goto 9110
      else if (no .eq. 25) then
*     SB-T2
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.050
      goto 9110
      else if (no .eq. 26) then
*     SB-T3
      coef_a = 13.71
      coef_b = 6980.0
      coef_c = 141375.0
      goto 9110
      else if (no .eq. 27) then
*     KHL-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.075
      goto 9110
      else if (no .eq. 28) then
*     KHL-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.080
      goto 9110
      else if (no .eq. 29) then
*     KHL-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.085
      goto 9110
      else if (no .eq. 30) then
*     SNR-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.090
      goto 9110
      else if (no .eq. 31) then
*     SNR-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.095
      goto 9110
      else if (no .eq. 32) then
*     SNR-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.100
      goto 9110
      else if (no .eq. 33) then
*     SNR-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.105
      goto 9110
      else if (no .eq. 34) then
*     TN-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.110
      goto 9110
      else if (no .eq. 35) then
*     TN-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.115
      goto 9110
      else if (no .eq. 36) then
*     RB-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.120
      goto 9110
      else if (no .eq. 37) then
*     RB-T2
```

```
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.125
      goto 9110
      else if (no .eq. 38) then
*     THB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.130
      goto 9110
      else if (no .eq. 39) then
*     THB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.135
      goto 9110
      else if (no .eq. 40) then
*     GCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.140

      goto 9110
      else if (no .eq. 41) then
*     HH-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.145
      goto 9110
      else if (no .eq. 42) then
*     HH-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.150
      goto 9110
      else if (no .eq. 43) then
*     AEP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.155
      goto 9110
      else if (no .eq. 44) then
*     TPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.160
      goto 9110
      else if (no .eq. 45) then
*     TTP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.165
      goto 9110
      else if (no .eq. 46) then
*     NPC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.170
      goto 9110
      else if (no .eq. 47) then
*     IP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.175
      goto 9110
      else if (no .eq. 48) then
*     IP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.180
      goto 9110
      else if (no .eq. 49) then
*     BCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.185
      goto 9110
      else if (no .eq. 50) then
*     MTP-T1
      coef_a = 0.0
```

```
      coef_b = 0.01
      coef_c = 0.190
      goto 9110
      else if (no .eq. 51) then
*     MTP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.195
      goto 9110
      else if (no .eq. 52) then
*     TCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.200
      goto 9110
      else if (no .eq. 53) then
*     TCC-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.205
      goto 9110
      else if (no .eq. 54) then
*     C0C0-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.210
      goto 9110
      else if (no .eq. 55) then
*     C0C0-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.215
      goto 9110
      else if (no .eq. 56) then
*     NPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 57) then
*     NPS-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 58) then
*     NPO-C2
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
      else if (no .eq. 59) then
*     WN-C3
      coef_a = -5.55
      coef_b = 7455.0
      coef_c = 50631.0
      goto 9110
      else if (no .eq. 60) then
*     KN-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.230
      goto 9110
      else if (no .eq. 61) then
*     RY-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.235
      goto 9110
      else if (no .eq. 62) then
*     RY-C4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.240
      goto 9110
      else if (no .eq. 63) then
*     BB-H5
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.245
```

```
      goto 9110
      else if (no .eq. 64) then
*     BB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.250
      goto 9110
      else if (no .eq. 65) then
*     BB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.255
      goto 9110
      else if (no .eq. 66) then
*     BLCP-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.260
      goto 9110
      else if (no .eq. 67) then
*     BLCP-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.265
      goto 9110
      else if (no .eq. 68) then
*     EPEC-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.270
      goto 9110
      else if (no .eq. 69) then
*     GULF-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.275
      goto 9110
      else if (no .eq. 70) then
*     GULF-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.280
      goto 9110
      end if

 9110   opfactor = megawt/mwhi*100.0
       heatrate = gcost * trfactor *
     +    (coef_a*opfactor**2 + coef_b*opfactor + coef_c)
     +    / 1000000.0

 9010 return
      end

      function indexv(dcval, szdcval, szarr, value)
*
*     Sorts discrete array. returns 0 if the value is not match.
*
      implicit none
      integer indexv, szdcval, szarr, i, expect, rangel, rangeh
      real dcval(1:szdcval), value

      indexv = 0
      if (value .lt. dcval(1)) goto 9015
      if (value .gt. dcval(szarr)) goto 9015
      expect = int((value - dcval(1))/(dcval(2)-dcval(1)))
      rangel = expect - 3
      rangeh = expect + 3
      if (rangel .lt. 1) then
        rangel = 1
      else if (rangeh .gt. szarr) then
        rangeh = szarr
      end if
      do 9020 i = rangel, rangeh
        if (dcval(i) .eq. value) then
           indexv = i
           goto 9015
        end if
 9020 continue
 9015 return
      end
```

ค.2 วิธีเจเนติกอัลกอริทึมร่วมกับตัวปฏิบัติการทางพันธุศาสตร์ที่มีประสิทธิภาพสูง

```
      program dedaoga
*
*     This program performs economic dispatch algorithms
*     for economic generation dispatch problems.
*
      implicit none

      integer maxgen, maxiter, iter, szpop, szchr, szind, szchild
      integer pop, popc, bestind, stamax, maxtrial, trapped, trapmax
      integer idum, ichr, ichild, i, j, itry, ic, iv
      integer m, n, ii, jj, iii, hrfn, iperiod, itmp, damaxgen
      integer mate, sztour, ipick, station, intval, chrtmp
      real k1, k2, k3, k4, rtmp, c
      real bestfit, bestfval, bestx, fitness, fitnessn
      real poppow, popcx, etime, elapsed(2), ltime, total
      real pcross, pmumax, fitmax, fitave, fitdev
      real bestfvalt, worstfvalt, bestxt, worstxt, fvaltave
      real besttt, worsttt, ttave
      real bestit, worstit, itave
      real losstmp, tmp, bias, biasac, pmutate
      real bcoeff, b0, b00, sumpow, heatrate
      real sumcost, costtmp, sumpowtmp, mustrunmw
      real mwnow, summw, sysloss, lowlim, hilim, mwmin, mwmax
      real rampdn, rampup, osysloss, osummw, edemand, odemand
      real coef_b, coef_c, tempb, Pr1, Pr2, remain
      real opmax, opmin, pmax, pmin, loadf
      logical forceup, forcedn, debug
      double precision val
      double precision fval, fvaltmp

      parameter (maxgen = 70, damaxgen = 15)
      parameter (szchild = 2, sztour = 4, intval = 3)
      parameter (szchr = 10, szind = (damaxgen-1)*szchr*intval)
      parameter (szpop = szind*2)

      dimension mate(szchild)
      dimension pop(szpop, szind), popc(szpop, szind)
      dimension poppow(szpop, maxgen*intval)
      dimension popcx(szpop, maxgen*intval)
      dimension fitness(szpop), fitnessn(szpop)
      dimension bestind(szind), bestx(maxgen*intval)
      dimension bestxt(maxgen*intval), worstxt(maxgen*intval)
      dimension mwnow(maxgen), lowlim(maxgen), hilim(maxgen)
      dimension mwmin(maxgen), mwmax(maxgen)
      dimension hrfn(maxgen), losstmp(maxgen), tempb(maxgen-1)
      dimension bcoeff(maxgen, maxgen), b0(maxgen)

      dimension loadf(intval), chrtmp(szchr)
      dimension rampdn(maxgen), rampup(maxgen)

*      data mwnow/maxgen*400.0/
*      data rampup/maxgen*50.0/
*      data rampdn/maxgen*80.0/

       mwnow(1) = 70.0
       mwnow(2) = 79.0
       mwnow(3) = 79.0
       mwnow(4) = 79.0
       mwnow(5) = 150.0
       mwnow(6) = 150.0
       mwnow(7) = 90.0
       mwnow(8) = 90.0
       mwnow(9) = 90.0
       mwnow(10) = 90.0
       mwnow(11) = 112.0
       mwnow(12) = 111.0
       mwnow(13) = 102.0
       mwnow(14) = 152.0
       mwnow(15) = 115.0
       mwnow(16) = 76.0
       mwnow(17) = 76.0
       mwnow(18) = 76.0
       mwnow(19) = 180.0
```

```
mwnow(20) = 700.0
mwnow(21) = 320.0
mwnow(22) = 693.0
mwnow(23) = 284.0
mwnow(24) = 140.0
mwnow(25) = 140.0
mwnow(26) = 140.0
mwnow(27) = 88.0
mwnow(28) = 88.0
mwnow(29) = 88.0
mwnow(30) = 120.0
mwnow(31) = 120.0
mwnow(32) = 120.0
mwnow(33) = 180.0
mwnow(34) = 20.0
mwnow(35) = 20.0
mwnow(36) = 700.0
mwnow(37) = 700.0
mwnow(38) = 105.0
mwnow(39) = 105.0
mwnow(40) = 90.0
mwnow(41) = 63.0
mwnow(42) = 63.0
mwnow(43) = 90.0
mwnow(44) = 25.0
mwnow(45) = 10.0
mwnow(46) = 32.0
mwnow(47) = 55.0
mwnow(48) = 55.0
mwnow(49) = 90.0
mwnow(50) = 60.0
mwnow(51) = 60.0
mwnow(52) = 90.0
mwnow(53) = 90.0
mwnow(54) = 90.0
mwnow(55) = 90.0
mwnow(56) = 90.0
mwnow(57) = 90.0
mwnow(58) = 180.0
mwnow(59) = 250.0
mwnow(60) = 70.0
mwnow(61) = 292.0
mwnow(62) = 292.0
mwnow(63) = 76.0
mwnow(64) = 76.0
mwnow(65) = 76.0
mwnow(66) = 673.0
mwnow(67) = 673.0
mwnow(68) = 350.0
mwnow(69) = 350.0
mwnow(70) = 350.0

rampup(1) = 5.0*6.8
rampup(2) = 5.0*40.0
rampup(3) = 5.0*40.0
rampup(4) = 5.0*40.0
rampup(5) = 5.0*5.0
rampup(6) = 5.0*5.0
rampup(7) = 5.0*3.0
rampup(8) = 5.0*3.0
rampup(9) = 5.0*3.0
rampup(10) = 5.0*3.0
rampup(11) = 5.0*70.0
rampup(12) = 5.0*70.0
rampup(13) = 5.0*70.0
rampup(14) = 5.0*120.0
rampup(15) = 5.0*115.0
rampup(16) = 5.0*76.3
rampup(17) = 5.0*76.3
rampup(18) = 5.0*76.3
rampup(19) = 5.0*8.0
rampup(20) = 5.0*39.0
rampup(21) = 5.0*22.5
rampup(22) = 5.0*25.0
rampup(23) = 5.0*10.0
rampup(24) = 5.0*35.0
rampup(25) = 5.0*35.0
rampup(26) = 5.0*35.0
rampup(27) = 5.0*87.0
rampup(28) = 5.0*87.0
```

```
rampup(29) = 5.0*87.0
rampup(30) = 5.0*70.2
rampup(31) = 5.0*70.2
rampup(32) = 5.0*70.2
rampup(33) = 5.0*70.2
rampup(34) = 5.0*20.0
rampup(35) = 5.0*20.0
rampup(36) = 5.0*39.0
rampup(37) = 5.0*39.0
rampup(38) = 5.0*70.0
rampup(39) = 5.0*70.0
rampup(40) = 5.0*35.0
rampup(41) = 5.0*63.0
rampup(42) = 5.0*63.0
rampup(43) = 5.0*35.0
rampup(44) = 5.0*35.0
rampup(45) = 5.0*10.0
rampup(46) = 5.0*35.0
rampup(47) = 5.0*35.0
rampup(48) = 5.0*35.0
rampup(49) = 5.0*35.0
rampup(50) = 5.0*35.0
rampup(51) = 5.0*35.0
rampup(52) = 5.0*35.0
rampup(53) = 5.0*35.0
rampup(54) = 5.0*35.0
rampup(55) = 5.0*35.0
rampup(56) = 5.0*35.0
rampup(57) = 5.0*35.0
rampup(58) = 5.0*8.0
rampup(59) = 5.0*39.0
rampup(60) = 5.0*6.8
rampup(61) = 5.0*55.0
rampup(62) = 5.0*55.0
rampup(63) = 5.0*76.3
rampup(64) = 5.0*76.3
rampup(65) = 5.0*76.3
rampup(66) = 5.0*10.0
rampup(67) = 5.0*10.0
rampup(68) = 5.0*8.0
rampup(69) = 5.0*8.0
rampup(70) = 5.0*8.0

rampdn(1) = 5.0*6.8
rampdn(2) = 5.0*40.0
rampdn(3) = 5.0*40.0
rampdn(4) = 5.0*40.0
rampdn(5) = 5.0*5.0
rampdn(6) = 5.0*5.0
rampdn(7) = 5.0*3.0
rampdn(8) = 5.0*3.0
rampdn(9) = 5.0*3.0
rampdn(10) = 5.0*3.0
rampdn(11) = 5.0*70.0
rampdn(12) = 5.0*70.0
rampdn(13) = 5.0*70.0
rampdn(14) = 5.0*120.0
rampdn(15) = 5.0*115.0
rampdn(16) = 5.0*76.3
rampdn(17) = 5.0*76.3
rampdn(18) = 5.0*76.3
rampdn(19) = 5.0*8.0
rampdn(20) = 5.0*39.0
rampdn(21) = 5.0*22.5
rampdn(22) = 5.0*25.0
rampdn(23) = 5.0*10.0
rampdn(24) = 5.0*35.0
rampdn(25) = 5.0*35.0
rampdn(26) = 5.0*35.0
rampdn(27) = 5.0*87.0
rampdn(28) = 5.0*87.0
rampdn(29) = 5.0*87.0
rampdn(30) = 5.0*70.2
rampdn(31) = 5.0*70.2
rampdn(32) = 5.0*70.2
rampdn(33) = 5.0*70.2
rampdn(34) = 5.0*20.0
rampdn(35) = 5.0*20.0
rampdn(36) = 5.0*39.0
rampdn(37) = 5.0*39.0
```

```
        rampdn(38) = 5.0*70.0
        rampdn(39) = 5.0*70.0
        rampdn(40) = 5.0*35.0
        rampdn(41) = 5.0*63.0
        rampdn(42) = 5.0*63.0
        rampdn(43) = 5.0*35.0
        rampdn(44) = 5.0*35.0
        rampdn(45) = 5.0*10.0
        rampdn(46) = 5.0*35.0
        rampdn(47) = 5.0*35.0
        rampdn(48) = 5.0*35.0
        rampdn(49) = 5.0*35.0
        rampdn(50) = 5.0*35.0
        rampdn(51) = 5.0*35.0
        rampdn(52) = 5.0*35.0
        rampdn(53) = 5.0*35.0
        rampdn(54) = 5.0*35.0
        rampdn(55) = 5.0*35.0
        rampdn(56) = 5.0*35.0
        rampdn(57) = 5.0*35.0
        rampdn(58) = 5.0*8.0
        rampdn(59) = 5.0*39.0
        rampdn(60) = 5.0*6.8
        rampdn(61) = 5.0*55.0
        rampdn(62) = 5.0*55.0
        rampdn(63) = 5.0*76.3
        rampdn(64) = 5.0*76.3
        rampdn(65) = 5.0*76.3
        rampdn(66) = 5.0*10.0
        rampdn(67) = 5.0*10.0
        rampdn(68) = 5.0*8.0
        rampdn(69) = 5.0*8.0
        rampdn(70) = 5.0*8.0

      loadf(1) = 1.002
      loadf(2) = 1.005
      loadf(3) = 1.07
*      loadf(4) = 0.985
*      loadf(5) = 0.992
*      loadf(6) = 1.015

      open(unit = 1, file = 'b.dat', status = 'old')

      do 1 i = 1, maxgen
        do 2 j = 1, maxgen
          read (1, *) bcoeff(i, j)
    2    continue
    1 continue
      close(1)

      open(unit = 2, file = 'b0.dat', status = 'old')
      do 3 i = 1, maxgen
        read (2, *) b0(i)
    3 continue
      read(2,*) b00
      close(2)

      lowlim(1) = 60.0
      lowlim(2) = 50.0
      lowlim(3) = 50.0

      lowlim(4) = 50.0
      lowlim(5) = 150.0
      lowlim(6) = 150.0
      lowlim(7) = 90.0
      lowlim(8) = 90.0
      lowlim(9) = 90.0
      lowlim(10) = 90.0
      lowlim(11) = 60.0
      lowlim(12) = 60.0
      lowlim(13) = 60.0
      lowlim(14) = 60.0
      lowlim(15) = 60.0
      lowlim(16) = 40.0
      lowlim(17) = 40.0
      lowlim(18) = 40.0
      lowlim(19) = 180.0
      lowlim(20) = 350.0
      lowlim(21) = 315.0
      lowlim(22) = 280.0
```

```
lowlim(23) = 284.0
lowlim(24) = 140.0
lowlim(25) = 140.0
lowlim(26) = 140.0
lowlim(27) = 50.0
lowlim(28) = 50.0
lowlim(29) = 50.0
lowlim(30) = 70.0
lowlim(31) = 70.0
lowlim(32) = 70.0
lowlim(33) = 120.0
lowlim(34) = 15.0
lowlim(35) = 15.0
lowlim(36) = 250.0
lowlim(37) = 250.0
lowlim(38) = 20.0
lowlim(39) = 20.0
lowlim(40) = 59.0
lowlim(41) = 10.0
lowlim(42) = 10.0
lowlim(43) = 59.0
lowlim(44) = 17.0
lowlim(45) = 5.0
lowlim(46) = 21.0
lowlim(47) = 36.0
lowlim(48) = 36.0
lowlim(49) = 59.0
lowlim(50) = 39.0
lowlim(51) = 39.0
lowlim(52) = 59.0
lowlim(53) = 59.0
lowlim(54) = 59.0
lowlim(55) = 59.0
lowlim(56) = 59.0
lowlim(57) = 59.0
lowlim(58) = 180.0
lowlim(59) = 250.0
lowlim(60) = 60.0
lowlim(61) = 250.0
lowlim(62) = 250.0
lowlim(63) = 40.0
lowlim(64) = 40.0
lowlim(65) = 40.0
lowlim(66) = 200.0
lowlim(67) = 200.0
lowlim(68) = 100.0
lowlim(69) = 100.0
lowlim(70) = 100.0

hilim(1) = 70.0
hilim(2) = 79.0
hilim(3) = 79.0
hilim(4) = 79.0
hilim(5) = 276.0
hilim(6) = 276.0
hilim(7) = 140.0
hilim(8) = 140.0
hilim(9) = 140.0
hilim(10) = 140.0
hilim(11) = 112.0
hilim(12) = 111.0
hilim(13) = 102.0
hilim(14) = 152.0
hilim(15) = 115.0
hilim(16) = 76.0
hilim(17) = 76.0
hilim(18) = 76.0
hilim(19) = 330.0
hilim(20) = 700.0
hilim(21) = 360.0
hilim(22) = 700.0
hilim(23) = 316.0
hilim(24) = 735.0
hilim(25) = 735.0
hilim(26) = 735.0
hilim(27) = 88.0
hilim(28) = 88.0
hilim(29) = 88.0
hilim(30) = 120.0
hilim(31) = 120.0
```

```
      hilim(32) = 120.0
      hilim(33) = 180.0
      hilim(34) = 20.0
      hilim(35) = 20.0
      hilim(36) = 700.0
      hilim(37) = 700.0
      hilim(38) = 105.0
      hilim(39) = 105.0
      hilim(40) = 90.0
      hilim(41) = 63.0
      hilim(42) = 63.0
      hilim(43) = 90.0
      hilim(44) = 25.0
      hilim(45) = 10.0
      hilim(46) = 32.0
      hilim(47) = 55.0
      hilim(48) = 55.0
      hilim(49) = 90.0
      hilim(50) = 60.0
      hilim(51) = 60.0
      hilim(52) = 90.0
      hilim(53) = 90.0
      hilim(54) = 90.0
      hilim(55) = 90.0
      hilim(56) = 90.0
      hilim(57) = 90.0
      hilim(58) = 330.0
      hilim(59) = 700.0
      hilim(60) = 70.0
      hilim(61) = 292.0
      hilim(62) = 292.0
      hilim(63) = 76.0
      hilim(64) = 76.0
      hilim(65) = 76.0
      hilim(66) = 673.0
      hilim(67) = 673.0
      hilim(68) = 350.0
      hilim(69) = 350.0
      hilim(70) = 350.0

      do 4 i = 1, maxgen
         hrfn(i) = i
         tmp = mwnow(i) - rampdn(i)
         call chkmw(tmp, lowlim(i), hilim(i))
         mwmin(i) = tmp
         tmp = mwnow(i) + rampup(i)
         call chkmw(tmp, lowlim(i), hilim(i))
         mwmax(i) = tmp
*        print *,
*     + mwmin(i), mwnow(i), mwmax(i)

    4 continue
*
*     Determining the system loss.
*         System loss = PiBijPj + Bi0Pi + B00
*
            sysloss = 0.0
            summw = 0.0
            sumcost = 0.0
            do 6140 m = 1, maxgen
               losstmp(m) = 0.0
               summw = summw + mwnow(m)
               sumcost = sumcost + heatrate(hrfn(m),
     + mwnow(m), lowlim(m), hilim(m))
               do 6145 n = 1, maxgen
                  losstmp(m) = losstmp(m) +
     +                    mwnow(n)*bcoeff(n,m)/100.0
 6145          continue
               sysloss = sysloss + losstmp(m)*mwnow(m)/100.0
 6140       continue

            do 6150 m = 1, maxgen
               sysloss = sysloss + mwnow(m)*b0(m)/100.0
 6150       continue
            sysloss = (sysloss + b00)*100.0
            odemand = summw - sysloss
            osysloss = sysloss
            osummw = summw
            print *, 'System test :  '
            print *, 'odemand = ', odemand,
```

```
     +      'osummw = ', osummw, 'osysloss (%) = ',
     +       osysloss/osummw*100.0
*           pause
*
*     Normalizing participation values
*         partcp = 0.xxxxx
*                    |---|
*                   nchrlen
*
      edemand = loadf(1)*odemand
      print *, 'Demand = ', edemand
      maxtrial = 100
      stamax = 100
      maxiter = 1000
      k1 = 1.0
      k2 = 0.5
      k3 = 1.0
      k4 = 0.5
      idum = -135000
      pmumax = 0.15
      biasac = 2.0
      trapmax = 4

*     *******************************************************
*                     Program Start
*     *******************************************************
      open(unit = 3, file = 'ogad.m', status = 'new')
      open(unit = 4, file = 'fitgad.m', status = 'new')
      open(unit = 5, file = 'ctgad.m', status = 'new')

      bestfvalt = 100000000000.0
      worstfvalt = 0.0
      fvaltave = 0.0
      besttt = 100000000.0
      worsttt = 0.00000001
      ttave = 0.0
      bestit = 100000000.0
      worstit = 1.0
      itave = 0.0
      write (3, *) 'oga = ['
      write (5, *) 'ct = ['

      do 9911 itry = 1, maxtrial
      print *, '###### Trial : ', itry, ' #######'
      write (4, *) 'fit', itry, ' = ['

      ic = 0
      iv = 0
      total = etime(elapsed)
      ltime = elapsed(1)
      bestfit = 0.0
      ipick = szpop
      station = 0
*     Initial random number generator
      call ran3(idum, val)
      do 5 i = 1, szpop
         do 10 j = 1, szind
            pop(i, j) = 1
            call ran3(1, val)
            if (val .lt. 0.5) pop(i, j) = 0
   10    continue
    5 continue

      do 920 iter = 1, maxiter
*        print *, '###### Generation : ', iter, ' #######'
         fitave = 0.0
         fitdev = 0.0
         fitmax = 0.0

         do 25 ichr = 1, szpop
*           print *, '###### Individual : ', ichr, ' #######'
            fval = 0.0
         do 1122 iperiod = 1, intval
*           print *, '###### Period : ', iperiod, ' #######'
            bias = 0.0
            trapped = 0

            if (iperiod .eq. 1) then
            do 8011 i = 1, maxgen
               tmp = mwnow(i) - rampdn(i)
```

```
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmin(i) = tmp
               tmp = mwnow(i) + rampup(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmax(i) = tmp
*              print *,
*     + mwmin(i), mwmax(i)
 8011       continue
            else
            do 8022 i = 1, maxgen
               tmp = poppow(ichr, ((iperiod-2)*(maxgen) + i))
     +         - rampdn(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmin(i) = tmp
               tmp = poppow(ichr, ((iperiod-2)*(maxgen) + i))
     +         + rampup(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmax(i) = tmp
*              print *,
*     + mwmin(i), poppow(ichr, ((iperiod-2)*(maxgen) + i)), mwmax(i)
 8022       continue
            end if

               opmax = 0.0
               opmin = 0.0
               mustrunmw = 0.0
               do 3331 j = 1, maxgen-1
                  if ((j .ge. 1) .and. (j .le. 4)) then
                     mustrunmw = mustrunmw + mwmax(j)
                     goto 3331
                  end if
                  if ((j .ge. 11) .and. (j .le. 18)) then
                     mustrunmw = mustrunmw + mwmax(j)
                     goto 3331
                  end if
                  if (j .eq. 20) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 3331
                  end if
                  if ((j .ge. 27) .and. (j .le. 57)) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 3331
                  end if
                  if ((j .ge. 60) .and. (j .le. 70)) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 3331
                  end if

                    opmax = opmax + mwmax(j)
                    opmin = opmin + mwmin(j)
 3331          continue

 2882       fvaltmp = 0.0
*           Check edemand
            pmax = opmax
            pmin = opmin
            remain = loadf(iperiod)*odemand+osysloss*loadf(iperiod)
     + -  (mwmax(maxgen)+mwmin(maxgen))/2.0 - mustrunmw
*            print *, 'remain', loadf(iperiod)*odemand,
*     + osysloss*loadf(iperiod),
*     + -(mwmax(maxgen)+mwmin(maxgen))/2.0, -mustrunmw
            sumpowtmp = 0.0
            forceup = .false.
            forcedn = .false.

            ii = 0
            do 30 i = 1, maxgen-1
*              Check dispatchable gernerating units
               if ((i .ge. 1) .and. (i .le. 4)) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                   goto 3333
               end if

               if ((i .ge. 11) .and. (i .le. 18)) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                   goto 3333
               end if

               if (i .eq. 20) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
```

```
                  goto 3333
               end if

               if ((i .ge. 27) .and. (i .le. 57)) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                  goto 3333
               end if

               if ((i .ge. 60) .and. (i .le. 70)) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                  goto 3333

               end if

               ii = ii + 1
               if (forceup) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                 do 653 m = 1, szchr
*                    Engraying...
                    if (m .eq. 1) then
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = 1
                    else
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = 0
                    end if
 653             continue
           remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
               pmax = pmax - mwmax(i)
               pmin = pmin - mwmin(i)

                 goto 3333
               end if

               if (forcedn) then
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
                 do 654 m = 1, szchr
*                    Engraying...
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +      = 0
 654             continue
           remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
               pmax = pmax - mwmax(i)
               pmin = pmin - mwmin(i)

                 goto 3333
               end if

          if (bias .ne. 0.0) then
*            print *, 'In first:', bias,
*     + poppow(ichr, ((iperiod-1)*(maxgen) + i))
             tmp = poppow(ichr, ((iperiod-1)*(maxgen) + i)) + bias

       if (tmp .gt. mwmax(i)) then
         bias = bias - mwmax(i) +
     +                poppow(ichr, ((iperiod-1)*(maxgen) + i))
           poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
       else if (tmp .lt. mwmin(i)) then
         bias = bias - mwmin(i) +
     +                poppow(ichr, ((iperiod-1)*(maxgen) + i))
           poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
       else
           poppow(ichr, ((iperiod-1)*(maxgen) + i)) = tmp
           bias = 0.0
       end if
*                   print *, 'At last:', bias,
*     + poppow(ichr, ((iperiod-1)*(maxgen) + i))
*                Re-encoding...
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 2991 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
        pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +            (szchr*(ii-1)+szchr+1-m))) = 1
                      tmp = tmp - 2.0**-m
                   else
        pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +            (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 2991            continue
```

```
*                Re-engraying...
                 do 2992 m = 2, szchr
       itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +      + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 2992             continue
                 do 2993 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 2993             continue
          else
               poppow(ichr, ((iperiod-1)*(maxgen) + i)) = 0.0
               itmp = 0
*               Degraying
               do 35 j = 1, szchr
            itmp = itmp + pop(ichr,
     + ((iperiod-1)*(damaxgen-1)*szchr+(szchr*(ii-1)+j)))
                 itmp = itmp-(itmp/2)*2
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + poppow(ichr, ((iperiod-1)*(maxgen) + i)) +
     + real(itmp)*2.0**-(szchr-j+1)
   35          continue
               poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + mwmin(i) + poppow(ichr, ((iperiod-1)*(maxgen) + i))*
     + (mwmax(i) - mwmin(i))
           end if

           remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)

*
*    ------------------------------------------------------
*
           if (ii .lt. damaxgen-1) then
              if (remain .ge. pmax) then
                 forceup = .true.
       remain = remain + poppow(ichr, ((iperiod-1)*(maxgen) + i))
       poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmax
*       if ((poppow(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (poppow(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     +  then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
*       print *, 'remain .ge. pmax ->', remain
       remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
*                Re-encoding...
*       print *, 'remain (new) ->', remain
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                 do 2901 m = 1, szchr
                    if (tmp .gt. 2.0**-m) then
          pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                 (szchr*(ii-1)+szchr+1-m))) = 1
                    tmp = tmp - 2.0**-m
                 else
          pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                 (szchr*(ii-1)+szchr+1-m))) = 0
                 end if
 2901            continue
*                Re-engraying...
                 do 2902 m = 2, szchr
       itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +  + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 2902             continue
                 do 2903 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 2903             continue
              end if
              if (remain .le. pmin) then
                 forcedn = .true.
       remain = remain + poppow(ichr, ((iperiod-1)*(maxgen) + i))
```

```
*       print *, 'In first:', poppow(ichr, ((iperiod-1)*(maxgen) + i))
*     + , mwmax(i), mwmin(i), remain
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmin
*       if ((poppow(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (poppow(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     + then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
          remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
*               Re-encoding...
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))

*          print *, 'check ', poppow(ichr, ((iperiod-1)*(maxgen) + i)),
*     +             mwmin(i), remain
               do 2904 m = 1, szchr
                  if (tmp .gt. 2.0**-m) then
        pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                    (szchr*(ii-1)+szchr+1-m))) = 1
                    tmp = tmp - 2.0**-m
                  else
        pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                    (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 2904             continue
*                 Re-engraying...
                 do 2905 m = 2, szchr
        itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) +
     +   pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 2905             continue
                 do 2906 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     + = chrtmp(m)
 2906             continue
              end if
           end if

*
*             Check power participation
*

 3333       sumpowtmp = sumpowtmp + poppow(ichr,
     + ((iperiod-1)*(maxgen) + i))

*               print *, 'Ind index = ', ichr, ' Gen = ', iter,
*     + ' Period = ', iperiod
*               print *, 'remain = ', remain, ' pmax = ',
*     + pmax, ' pmin = ', pmin
*               print *, 'Pgen', i, mwmin(i), '<=',
*     + poppow(ichr, ((iperiod-1)*(maxgen) + i)), '<=', mwmax(i)
*               print *, 'forceup = ', forceup, ' forcedn = ', forcedn
*               print *, 'sumpowtmp = ', sumpowtmp
*               pause

              costtmp = heatrate(hrfn(i),
     + poppow(ichr, ((iperiod-1)*(maxgen) + i)), lowlim(i),
     + hilim(i))
*              fval (MBaht/h)
               fvaltmp = fvaltmp + costtmp/1000000.0
*              print *, 'Total cost =', fvaltmp
   30       continue
*
*     Determine Pr
*          Demand - SumMW + Ploss - Pr = 0
*          Ploss = aPr^2 + bPr + c
*              a = Brr
*              b = BraPa + PatBar + B0r
*              c = PatBaaPa + PatB0a + B00
*          Thus, modify coefficients.
*              a = Brr
*              b = b-1 = BraPa + PatBar + B0r - 1
*              c = PatBaaPa + PatB0a + B00 + Demand - sumMW
*          Determines Pr
*             Pr = -b +- sqrt(b^2 - 4*a*c)
```

```
*                  ----------------------
*                           2*a

           coef_b = 0.0
           coef_c = 0.0

           do 8903 ii = 1, maxgen-1
*         print *, 'u', ii, poppow(ichr, ((iperiod-1)*(maxgen-1) + ii))
              coef_b = coef_b +
     + bcoeff(maxgen, ii)*poppow(ichr, ((iperiod-1)*(maxgen-1) +
     + ii))/100.0
              tempb(ii) = 0.0
 8903       continue
           coef_b = coef_b*2.0 + b0(maxgen) - 1.0

           do 8904 ii = 1, maxgen-1
              do 8905 iii = 1, maxgen-1
                 tempb(ii) = tempb(ii) +
     + poppow(ichr, ((iperiod-1)*(maxgen)+iii))*bcoeff(iii, ii)/100.0
 8905         continue
 8904       continue
           do 8906 ii = 1, maxgen-1
              coef_c = coef_c + tempb(ii)*
     + poppow(ichr, ((iperiod-1)*(maxgen) + ii))/100.0
 8906       continue
           do 8907 ii = 1, maxgen-1
              coef_c = coef_c + poppow(ichr,((iperiod-1)*(maxgen)+ii))
     +        *b0(ii)/100.0
 8907       continue
       coef_c = coef_c + b00 + (loadf(iperiod)*odemand-sumpowtmp)/100.0
*           print *, 'load', loadf(iperiod)*odemand, sumpowtmp
           Pr1 = -1.0*coef_b + sqrt(coef_b*coef_b -
     +                          4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr1 = 100.0*Pr1/(2.0*bcoeff(maxgen,maxgen))
           Pr2 = -1.0*coef_b - sqrt(coef_b*coef_b -
     +                          4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr2 = 100.0*Pr2/(2.0*bcoeff(maxgen,maxgen))

           if (Pr1 .le. 0.0) then
              if (Pr2 .gt. 0.0) Pr1 = Pr2
           end if
           if (Pr1 .gt. Pr2) Pr1 = Pr2
           sumpowtmp = sumpowtmp + Pr1
           poppow(ichr, (iperiod-1)*maxgen + maxgen) = Pr1

           if (Pr1 .gt. mwmax(maxgen)) then
*             print *, 'Power imbalance occurred. (Pr1 > mwmax)',
*     +             Pr1, '>', mwmax(maxgen),
*     +      '(IChr = ', ichr, ' Period = ', iperiod, ')'
            if (trapped .gt. trapmax) then
              print *, 'Trapped:', trapped
*             Re-building the chromosome
              do 7090 ii = 1, szchr*(damaxgen-1)
                 pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +           + ii)) = 1
                 call ran3(1, val)
                 if (val .lt. 0.5) then
                    pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +           + ii)) = 0
                  end if
 7090         continue
             trapped = 0
             goto 2882
            end if
              bias = Pr1 - mwmax(maxgen)
*             Loss compensating...
              bias = bias + bias*sysloss/sumpowtmp*biasac
              trapped = trapped + 1
              goto 2882
           end if
            if (Pr1 .lt. mwmin(maxgen)) then
*              print *, 'Power imbalance occurred. (Pr1 < mwmin)',
*     +      Pr1, '<', mwmin(maxgen),
*     +      '(IChr = ', ichr, ' Period = ', iperiod, ')'

             if (trapped .gt. trapmax) then
               print *, 'Trapped:', trapped
*              Re-building the chromosome
               do 7091 ii = 1, szchr*(damaxgen-1)
                  pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
```

```
     +          + ii)) = 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 0
                end if
7091          continue
              trapped = 0
              goto 2882
            end if
              bias = Pr1 - mwmin(maxgen)
*             Loss compensating...
              bias = bias + bias*sysloss/sumpowtmp*biasac
              trapped = trapped + 1
              goto 2882
           end if
            costtmp = heatrate(hrfn(maxgen), Pr1, mwmin(maxgen),
     + mwmax(maxgen))
           sumpow = sumpowtmp
*           fval (MBaht)
           fvaltmp = fvaltmp + costtmp/1000000.0
           fval = fval + fvaltmp
*          print *, 'fval = ', fval, costtmp
           bias = 0.0

*           If Pr limit is violated.

           if (costtmp .ge. 1.0e100) then
*               iv = iv + 1
*              print *, 'Pr limit violated...', Pr1
           else
*               ic = ic + 1
*              print *, 'Cool !... '
           end if
 1122  continue
*
*            Evaluating fitness values
*
            fitness(ichr) = 1.0 / fval
            fitave = fitave + fitness(ichr)
            if (fitness(ichr) .gt. fitmax) then
               fitmax = fitness(ichr)
            end if

            if (fitness(ichr) .gt. bestfit) then
*              print *, fval
               station = 0
               bestfit = fitness(ichr)
               bestfval = fval
               do 55 m = 1, szind
                  bestind(m) = pop(ichr, m)
   55           continue
               do 60 n = 1, maxgen*intval
                  bestx(n) = poppow(ichr, n)
   60           continue
            end if
   25     continue

*         End ichr

          fitave = fitave / real(szpop)
          write (4, 666) fitave, bestfit
  666     format (f15.5, f15.5)

*          print *, 'Fitness scaling ...'
*
*         -------------  Sigma Truncation Scaling --------------
*         f = f - (fave - c*sigma)
*         The constant c is chosen as a reasonable muliple of the
*         population standard deviation (between 1 and 3) and
*         negative results are arbitrarily set to 0.
*
          c = 2.0
          do 61 i = 1, szpop
             fitdev = fitdev + (fitness(i) - fitave)**2
   61     continue
          fitdev = fitdev**0.5

          do 62 i = 1, szpop
             fitnessn(i) = fitness(i) - fitave + c*fitdev
```

```
            if (fitnessn(i) .lt. 0.0) fitness(i) = 0.0
   62     continue

*
*         Performing Tournament selection.
*
          do 65 ichr = 1, szpop, szchild
             do 70 ichild = 1, szchild
                if ((ipick+sztour) .gt. szpop) then
*                   Shuffles population randomly.
                  ipick = 1

                  do 75 m = 1, szpop-1
                     call ran3(1, val)
                     i = m + dint(dble(szpop-m)*val) + 1
                     do 80 n = 1, maxgen
                        j = pop(i, n)
                        pop(i, n) = pop(m, n)
                        pop(m, n) = j
   80                 continue
                     rtmp = fitness(i)
                     fitness(i) = fitness(m)
                     fitness(m) = rtmp
                     rtmp = fitnessn(i)
                     fitnessn(i) = fitnessn(m)
                     fitnessn(m) = rtmp
   75              continue
                end if

                mate(ichild) = ipick
                do 85 i = 1, sztour-1
                   if (fitnessn(mate(ichild)) .lt. fitnessn(ipick+i))
     +              then
                      mate(ichild) = ipick + i
                   end if
   85             continue
                ipick = ipick + sztour
   70        continue
*
*            Creates child chromosomes.
*
             do 90 n = 1, szind
                popc(ichr, n) = pop(mate(1), n)
                popc(ichr+1, n) = pop(mate(2), n)
   90        continue
*
*            Performing uniform crossover.
*
*             print *, 'Crossing over ...'
             if (fitness(mate(1)) .gt. fitness(mate(2))) then
                if (fitness(mate(1)) .ge. fitave) then
                   pcross = k1*(fitmax-fitness(mate(1)))/(fitmax-fitave)
                else
                   pcross = k3
                end if
             else
                if (fitness(mate(2)) .ge. fitave) then
                   pcross = k1*(fitmax-fitness(mate(2)))/(fitmax-fitave)
                else
                   pcross = k3
                end if
             end if
             do 95 n = 1, szind
                call ran3(1, val)
                if (val .lt. pcross) then
                  popc(ichr, n) = pop(mate(2), n)
                  popc(ichr+1, n) = pop(mate(1), n)
                end if
   95        continue

*         End ichr (selection)
*
*         Performing random mutation.
*
          print *, 'Mutating ...'
*
*         Performing Jump mutation.
*
          pmutate = real(station)/real(stamax)*pmumax
          call ran3(1, val)
```

```
         if (val .lt. pmutate) then
            do 6006 iperiod = 1, intval
              ii = 0
              do 6007 i = 1, maxgen-1
               if ((i .ge. 1) .and. (i .le. 4)) then
                   goto 6007
                  end if

               if ((i .ge. 11) .and. (i .le. 18)) then
                   goto 6007
               end if

               if (i .eq. 20) then
                   goto 6007
               end if

               if ((i .ge. 27) .and. (i .le. 57)) then
                   goto 6007
               end if

               if ((i .ge. 60) .and. (i .le. 70)) then
                   goto 6007
               end if

                 ii = ii + 1
                 call ran3(1, val)
                 if (val .lt. 0.5) then
                    popcx(ichr, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr, ((iperiod-1)*(maxgen) + i)) -
     + 2**-szchr*(mwmax(i) - mwmin(i))
                 else
                   popcx(ichr, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr, ((iperiod-1)*(maxgen) + i)) +
     + 2**-szchr*(mwmax(i) - mwmin(i))
                 end if
          tmp = (popcx(ichr, ((iperiod-1)*(maxgen) + i))- mwmin(i))/
     + (mwmax(i) - mwmin(i))
                 do 6008 m = 1, szchr
                    if (tmp .gt. 2.0**-m) then
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                      (szchr*(ii-1)+szchr+1-m))) = 1
                       tmp = tmp - 2.0**-m
                    else
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                      (szchr*(ii-1)+szchr+1-m))) = 0
                    end if
 6008            continue
*                Re-engraying...
                 do 6009 m = 2, szchr
        itmp = popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +  + (szchr*(ii-1)+m-1)))

                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 6009             continue
                 do 6010 m = 2, szchr
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 6010            continue
 6007         continue
 6006      continue
         end if

          call ran3(1, val)
          if (val .lt. pmutate) then
            do 6012 iperiod = 1, intval
              ii = 0
              do 6013 i = 1, maxgen-1
               if ((i .ge. 1) .and. (i .le. 4)) then
                   goto 6013
                  end if

               if ((i .ge. 11) .and. (i .le. 18)) then
                   goto 6013
               end if

               if (i .eq. 20) then
                   goto 6013
               end if
```

```
              if ((i .ge. 27) .and. (i .le. 57)) then
                  goto 6013
              end if

              if ((i .ge. 60) .and. (i .le. 70)) then
                  goto 6013
              end if

                ii = ii + 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) -
     + 2**-szchr*(mwmax(i) - mwmin(i))
                else
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) +
     + 2**-szchr*(mwmax(i) - mwmin(i))
                end if
         tmp = (popcx(ichr+1,((iperiod-1)*(maxgen) + i)) -mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 6014 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 1
                      tmp = tmp - 2.0**-m
                   else
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 6014             continue
*                 Re-engraying...
                 do 6015 m = 2, szchr
          itmp = popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr+1,((iperiod-1)*(damaxgen-1)
     + *szchr + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 6015             continue
                 do 6016 m = 2, szchr
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +     = chrtmp(m)
 6016             continue
 6013         continue
 6012      continue
        end if

  65    continue
*
*        Replace new generation with elitism.
*
*         print *, 'Regenerating...'
         i = 0
         do 110 m = 1, szpop
            j = 0
            do 115 n = 1, szind
               pop(m, n) = popc(m, n)
               if (pop(m, n) .eq. bestind(n)) then
                  j = j + 1
               end if
  115       continue
  110    continue
         if (j .ne. szind) then
            call ran3(1, val)
            i = 1 + dint(dble(szpop)*val)
            do 120 n = 1, szind
               pop(i, n) = bestind(n)
  120       continue
         end if
*
*        Checks population homogeneity
*
*         do 125 i = 1, szind
*            numzero(i) = 0
*  125    continue
*         tmp = 0.0
*         do 130 i = 1, szind
*            do 135 j = 1, szpop
*               if (pop(j, i) .eq. 0) then
```

```
*                  numzero(i) = numzero(i) + 1
*               end if
*  135       continue
*            if (numzero(i) .ge. (szpop/2)) then
*               tmp = tmp + real(numzero(i)) / real(szpop)
*            else
*               tmp = tmp + real(szpop - numzero(i)) / real(szpop)
*            end if
*  130    continue
*         tmp = tmp / real(szind)
*          print *, 'best heat = ', bestheat, ' and power diff = ',
*      + (bestgen - loadf(iperiod)*odemand - bestloss)
*          print *, '% Homogenity = ', tmp
*
*        Checks termination criterion
*
         station = station + 1
*         if ((tmp .gt. hmlevel) .or. (station .gt. 250)) goto 500
          if (station .gt. stamax) goto 500

  920  continue

*       End generation
*     print *, 'Generation complete !'
500     total = etime(elapsed)
        write (3, *) bestx(1), bestx(2), bestx(3), bestx(4),
     + bestx(5), bestx(6), bestx(7), bestx(8), bestx(9), bestx(10),
     + bestx(11), bestx(12), bestx(13), bestx(14), bestx(15),
     + bestx(16), bestx(17), bestx(18), bestx(19), bestx(20),
     + bestx(21), bestx(22), bestx(23), bestx(24), bestx(25),
     + bestx(26), bestx(27), bestx(28), bestx(29), bestx(30),
     + bestx(31), bestx(32), bestx(33), bestx(34), bestx(35),
     + bestx(36), bestx(37), bestx(38), bestx(39), bestx(40),
     + bestx(41), bestx(42), bestx(43), bestx(44), bestx(45),
     + bestx(46), bestx(47), bestx(48), bestx(49), bestx(50),
     + bestx(51), bestx(52), bestx(53), bestx(54), bestx(55),
     + bestx(56), bestx(57), bestx(58), bestx(59), bestx(60),
     + bestx(61), bestx(62), bestx(63), bestx(64), bestx(65),
     + bestx(66), bestx(67), bestx(68), bestx(69), bestx(70),
     + bestx(71), bestx(72), bestx(73), bestx(74), bestx(75),
     + bestx(76), bestx(77), bestx(78), bestx(79), bestx(80),
     + bestx(81), bestx(82), bestx(83), bestx(84), bestx(85),
     + bestx(86), bestx(87), bestx(88), bestx(89), bestx(90),
     + bestx(91), bestx(92), bestx(93), bestx(94), bestx(95),
     + bestx(96), bestx(97), bestx(98), bestx(99), bestx(100),
     + bestx(101), bestx(102), bestx(103), bestx(104), bestx(105),
     + bestx(106), bestx(107), bestx(108), bestx(109), bestx(110),
     + bestx(111), bestx(112), bestx(113), bestx(114), bestx(115),
     + bestx(116), bestx(117), bestx(118), bestx(119), bestx(120),
     + bestx(121), bestx(122), bestx(123), bestx(124), bestx(125),
     + bestx(126), bestx(127), bestx(128), bestx(129), bestx(130),
     + bestx(131), bestx(132), bestx(133), bestx(134), bestx(135),
     + bestx(136), bestx(137), bestx(138), bestx(139), bestx(140),
     + bestx(141), bestx(142), bestx(143), bestx(144), bestx(145),
     + bestx(146), bestx(147), bestx(148), bestx(149), bestx(150),
     + bestx(151), bestx(152), bestx(153), bestx(154), bestx(155),
     + bestx(156), bestx(157), bestx(158), bestx(159), bestx(160),
     + bestx(161), bestx(162), bestx(163), bestx(164), bestx(165),
     + bestx(166), bestx(167), bestx(168), bestx(169), bestx(170),
     + bestx(171), bestx(172), bestx(173), bestx(174), bestx(175),
     + bestx(176), bestx(177), bestx(178), bestx(179), bestx(180),
     + bestx(181), bestx(182), bestx(183), bestx(184), bestx(185),
     + bestx(186), bestx(187), bestx(188), bestx(189), bestx(190),
     + bestx(191), bestx(192), bestx(193), bestx(194), bestx(195),
     + bestx(196), bestx(197), bestx(198), bestx(199), bestx(200),
     + bestx(201), bestx(202), bestx(203), bestx(204), bestx(205),
     + bestx(206), bestx(207), bestx(208), bestx(209), bestx(210),
     + bestfval
       write (5, *) bestfval, iter, elapsed(1)-ltime
        if (bestfval .lt. bestfvalt) then
            bestfvalt = bestfval
            bestxt(1) = bestx(1)
            bestxt(2) = bestx(2)
            bestxt(3) = bestx(3)
            bestxt(4) = bestx(4)
            bestxt(5) = bestx(5)
            bestxt(6) = bestx(6)
            bestxt(7) = bestx(7)
            bestxt(8) = bestx(8)
            bestxt(9) = bestx(9)
```

```
bestxt(10) = bestx(10)
bestxt(11) = bestx(11)
bestxt(12) = bestx(12)
bestxt(13) = bestx(13)
bestxt(14) = bestx(14)
bestxt(15) = bestx(15)
bestxt(16) = bestx(16)
bestxt(17) = bestx(17)
bestxt(18) = bestx(18)
bestxt(19) = bestx(19)
bestxt(20) = bestx(20)
bestxt(21) = bestx(21)
bestxt(22) = bestx(22)
bestxt(23) = bestx(23)
bestxt(24) = bestx(24)
bestxt(25) = bestx(25)
bestxt(26) = bestx(26)
bestxt(27) = bestx(27)
bestxt(28) = bestx(28)
bestxt(29) = bestx(29)
bestxt(30) = bestx(30)
bestxt(31) = bestx(31)
bestxt(32) = bestx(32)
bestxt(33) = bestx(33)
bestxt(34) = bestx(34)
bestxt(35) = bestx(35)
bestxt(36) = bestx(36)
bestxt(37) = bestx(37)
bestxt(38) = bestx(38)
bestxt(39) = bestx(39)
bestxt(40) = bestx(40)
bestxt(41) = bestx(41)
bestxt(42) = bestx(42)
bestxt(43) = bestx(43)
bestxt(44) = bestx(44)
bestxt(45) = bestx(45)
bestxt(46) = bestx(46)
bestxt(47) = bestx(47)
bestxt(48) = bestx(48)
bestxt(49) = bestx(49)
bestxt(50) = bestx(50)
bestxt(51) = bestx(51)
bestxt(52) = bestx(52)
bestxt(53) = bestx(53)
bestxt(54) = bestx(54)
bestxt(55) = bestx(55)
bestxt(56) = bestx(56)
bestxt(57) = bestx(57)
bestxt(58) = bestx(58)
bestxt(59) = bestx(59)
bestxt(60) = bestx(60)
bestxt(61) = bestx(61)
bestxt(62) = bestx(62)
bestxt(63) = bestx(63)
bestxt(64) = bestx(64)
bestxt(65) = bestx(65)
bestxt(66) = bestx(66)
bestxt(67) = bestx(67)
bestxt(68) = bestx(68)
bestxt(69) = bestx(69)
bestxt(70) = bestx(70)
bestxt(71) = bestx(71)
bestxt(72) = bestx(72)
bestxt(73) = bestx(73)
bestxt(74) = bestx(74)
bestxt(75) = bestx(75)
bestxt(76) = bestx(76)
bestxt(77) = bestx(77)
bestxt(78) = bestx(78)
bestxt(79) = bestx(79)
bestxt(80) = bestx(80)
bestxt(81) = bestx(81)
bestxt(82) = bestx(82)
bestxt(83) = bestx(83)
bestxt(84) = bestx(84)
bestxt(85) = bestx(85)
bestxt(86) = bestx(86)
bestxt(87) = bestx(87)
bestxt(88) = bestx(88)
bestxt(89) = bestx(89)
```

```
bestxt(90) = bestx(90)
bestxt(91) = bestx(91)
bestxt(92) = bestx(92)
bestxt(93) = bestx(93)
bestxt(94) = bestx(94)
bestxt(95) = bestx(95)
bestxt(96) = bestx(96)
bestxt(97) = bestx(97)
bestxt(98) = bestx(98)
bestxt(99) = bestx(99)
bestxt(100) = bestx(100)
bestxt(101) = bestx(101)
bestxt(102) = bestx(102)
bestxt(103) = bestx(103)
bestxt(104) = bestx(104)
bestxt(105) = bestx(105)
bestxt(106) = bestx(106)
bestxt(107) = bestx(107)
bestxt(108) = bestx(108)
bestxt(109) = bestx(109)
bestxt(110) = bestx(110)
bestxt(111) = bestx(111)
bestxt(112) = bestx(112)
bestxt(113) = bestx(113)
bestxt(114) = bestx(114)
bestxt(115) = bestx(115)
bestxt(116) = bestx(116)
bestxt(117) = bestx(117)
bestxt(118) = bestx(118)
bestxt(119) = bestx(119)
bestxt(120) = bestx(120)
bestxt(121) = bestx(121)
bestxt(122) = bestx(122)
bestxt(123) = bestx(123)
bestxt(124) = bestx(124)
bestxt(125) = bestx(125)
bestxt(126) = bestx(126)

bestxt(127) = bestx(127)
bestxt(128) = bestx(128)
bestxt(129) = bestx(129)
bestxt(130) = bestx(130)
bestxt(131) = bestx(131)
bestxt(132) = bestx(132)
bestxt(133) = bestx(133)
bestxt(134) = bestx(134)
bestxt(135) = bestx(135)
bestxt(136) = bestx(136)
bestxt(137) = bestx(137)
bestxt(138) = bestx(138)
bestxt(139) = bestx(139)
bestxt(140) = bestx(140)
bestxt(141) = bestx(141)
bestxt(142) = bestx(142)
bestxt(143) = bestx(143)
bestxt(144) = bestx(144)
bestxt(145) = bestx(145)
bestxt(146) = bestx(146)
bestxt(147) = bestx(147)
bestxt(148) = bestx(148)
bestxt(149) = bestx(149)
bestxt(150) = bestx(150)
bestxt(151) = bestx(151)
bestxt(152) = bestx(152)
bestxt(153) = bestx(153)
bestxt(154) = bestx(154)
bestxt(155) = bestx(155)
bestxt(156) = bestx(156)
bestxt(157) = bestx(157)
bestxt(158) = bestx(158)
bestxt(159) = bestx(159)
bestxt(160) = bestx(160)
bestxt(161) = bestx(161)
bestxt(162) = bestx(162)
bestxt(163) = bestx(163)
bestxt(164) = bestx(164)
bestxt(165) = bestx(165)
bestxt(166) = bestx(166)
bestxt(167) = bestx(167)
bestxt(168) = bestx(168)
```

```
      bestxt(169) = bestx(169)
      bestxt(170) = bestx(170)
      bestxt(171) = bestx(171)
      bestxt(172) = bestx(172)
      bestxt(173) = bestx(173)
      bestxt(174) = bestx(174)
      bestxt(175) = bestx(175)
      bestxt(176) = bestx(176)
      bestxt(177) = bestx(177)
      bestxt(178) = bestx(178)
      bestxt(179) = bestx(179)
      bestxt(180) = bestx(180)
      bestxt(181) = bestx(181)
      bestxt(182) = bestx(182)
      bestxt(183) = bestx(183)
      bestxt(184) = bestx(184)
      bestxt(185) = bestx(185)
      bestxt(186) = bestx(186)
      bestxt(187) = bestx(187)
      bestxt(188) = bestx(188)
      bestxt(189) = bestx(189)
      bestxt(190) = bestx(190)
      bestxt(191) = bestx(191)
      bestxt(192) = bestx(192)
      bestxt(193) = bestx(193)
      bestxt(194) = bestx(194)
      bestxt(195) = bestx(195)
      bestxt(196) = bestx(196)
      bestxt(197) = bestx(197)
      bestxt(198) = bestx(198)
      bestxt(199) = bestx(199)
      bestxt(200) = bestx(200)
      bestxt(201) = bestx(201)
      bestxt(202) = bestx(202)
      bestxt(203) = bestx(203)
      bestxt(204) = bestx(204)
      bestxt(205) = bestx(205)
      bestxt(206) = bestx(206)
      bestxt(207) = bestx(207)
      bestxt(208) = bestx(208)
      bestxt(209) = bestx(209)
      bestxt(210) = bestx(210)
    end if
    if (bestfval .gt. worstfvalt) then
        worstfvalt = bestfval
      worstxt(1)= bestx(1)
      worstxt(2)= bestx(2)
      worstxt(3)= bestx(3)
      worstxt(4)= bestx(4)
      worstxt(5)= bestx(5)
      worstxt(6)= bestx(6)
      worstxt(7)= bestx(7)
      worstxt(8)= bestx(8)
      worstxt(9)= bestx(9)
      worstxt(10)= bestx(10)
      worstxt(11)= bestx(11)
      worstxt(12)= bestx(12)
      worstxt(13)= bestx(13)
      worstxt(14)= bestx(14)
      worstxt(15)= bestx(15)
      worstxt(16)= bestx(16)
      worstxt(17)= bestx(17)
      worstxt(18)= bestx(18)
      worstxt(19)= bestx(19)
      worstxt(20)= bestx(20)
      worstxt(21)= bestx(21)
      worstxt(22)= bestx(22)
      worstxt(23)= bestx(23)
      worstxt(24)= bestx(24)
      worstxt(25)= bestx(25)
      worstxt(26)= bestx(26)
      worstxt(27)= bestx(27)
      worstxt(28)= bestx(28)
      worstxt(29)= bestx(29)
      worstxt(30)= bestx(30)
      worstxt(31)= bestx(31)
      worstxt(32)= bestx(32)
      worstxt(33)= bestx(33)
      worstxt(34)= bestx(34)
      worstxt(35)= bestx(35)
```

worstxt(36)= bestx(36)
worstxt(37)= bestx(37)
worstxt(38)= bestx(38)
worstxt(39)= bestx(39)
worstxt(40)= bestx(40)
worstxt(41)= bestx(41)
worstxt(42)= bestx(42)
worstxt(43)= bestx(43)
worstxt(44)= bestx(44)
worstxt(45)= bestx(45)
worstxt(46)= bestx(46)
worstxt(47)= bestx(47)
worstxt(48)= bestx(48)
worstxt(49)= bestx(49)
worstxt(50)= bestx(50)
worstxt(51)= bestx(51)
worstxt(52)= bestx(52)
worstxt(53)= bestx(53)
worstxt(54)= bestx(54)
worstxt(55)= bestx(55)
worstxt(56)= bestx(56)
worstxt(57)= bestx(57)
worstxt(58)= bestx(58)
worstxt(59)= bestx(59)
worstxt(60)= bestx(60)
worstxt(61)= bestx(61)
worstxt(62)= bestx(62)
worstxt(63)= bestx(63)
worstxt(64)= bestx(64)
worstxt(65)= bestx(65)
worstxt(66)= bestx(66)
worstxt(67)= bestx(67)
worstxt(68)= bestx(68)
worstxt(69)= bestx(69)
worstxt(70)= bestx(70)
worstxt(71)= bestx(71)
worstxt(72)= bestx(72)
worstxt(73)= bestx(73)
worstxt(74)= bestx(74)
worstxt(75)= bestx(75)
worstxt(76)= bestx(76)
worstxt(77)= bestx(77)
worstxt(78)= bestx(78)
worstxt(79)= bestx(79)
worstxt(80)= bestx(80)
worstxt(81)= bestx(81)
worstxt(82)= bestx(82)
worstxt(83)= bestx(83)
worstxt(84)= bestx(84)
worstxt(85)= bestx(85)
worstxt(86)= bestx(86)
worstxt(87)= bestx(87)
worstxt(88)= bestx(88)
worstxt(89)= bestx(89)
worstxt(90)= bestx(90)
worstxt(91)= bestx(91)
worstxt(92)= bestx(92)
worstxt(93)= bestx(93)
worstxt(94)= bestx(94)
worstxt(95)= bestx(95)
worstxt(96)= bestx(96)
worstxt(97)= bestx(97)
worstxt(98)= bestx(98)
worstxt(99)= bestx(99)
worstxt(100)= bestx(100)
worstxt(101)= bestx(101)
worstxt(102)= bestx(102)
worstxt(103)= bestx(103)
worstxt(104)= bestx(104)
worstxt(105)= bestx(105)
worstxt(106)= bestx(106)
worstxt(107)= bestx(107)
worstxt(108)= bestx(108)
worstxt(109)= bestx(109)
worstxt(110)= bestx(110)
worstxt(111)= bestx(111)
worstxt(112)= bestx(112)
worstxt(113)= bestx(113)
worstxt(114)= bestx(114)
worstxt(115)= bestx(115)

worstxt(116)= bestx(116)
worstxt(117)= bestx(117)
worstxt(118)= bestx(118)
worstxt(119)= bestx(119)
worstxt(120)= bestx(120)
worstxt(121)= bestx(121)
worstxt(122)= bestx(122)
worstxt(123)= bestx(123)
worstxt(124)= bestx(124)
worstxt(125)= bestx(125)
worstxt(126)= bestx(126)
worstxt(127)= bestx(127)
worstxt(128)= bestx(128)
worstxt(129)= bestx(129)
worstxt(130)= bestx(130)
worstxt(131)= bestx(131)
worstxt(132)= bestx(132)
worstxt(133)= bestx(133)
worstxt(134)= bestx(134)
worstxt(135)= bestx(135)
worstxt(136)= bestx(136)

worstxt(137)= bestx(137)
worstxt(138)= bestx(138)
worstxt(139)= bestx(139)
worstxt(140)= bestx(140)
worstxt(141)= bestx(141)
worstxt(142)= bestx(142)
worstxt(143)= bestx(143)
worstxt(144)= bestx(144)
worstxt(145)= bestx(145)
worstxt(146)= bestx(146)
worstxt(147)= bestx(147)
worstxt(148)= bestx(148)
worstxt(149)= bestx(149)
worstxt(150)= bestx(150)
worstxt(151)= bestx(151)
worstxt(152)= bestx(152)
worstxt(153)= bestx(153)
worstxt(154)= bestx(154)
worstxt(155)= bestx(155)
worstxt(156)= bestx(156)
worstxt(157)= bestx(157)
worstxt(158)= bestx(158)
worstxt(159)= bestx(159)
worstxt(160)= bestx(160)
worstxt(161)= bestx(161)
worstxt(162)= bestx(162)
worstxt(163)= bestx(163)
worstxt(164)= bestx(164)
worstxt(165)= bestx(165)
worstxt(166)= bestx(166)
worstxt(167)= bestx(167)
worstxt(168)= bestx(168)
worstxt(169)= bestx(169)
worstxt(170)= bestx(170)
worstxt(171)= bestx(171)
worstxt(172)= bestx(172)
worstxt(173)= bestx(173)
worstxt(174)= bestx(174)
worstxt(175)= bestx(175)
worstxt(176)= bestx(176)
worstxt(177)= bestx(177)
worstxt(178)= bestx(178)
worstxt(179)= bestx(179)
worstxt(180)= bestx(180)
worstxt(181)= bestx(181)
worstxt(182)= bestx(182)
worstxt(183)= bestx(183)
worstxt(184)= bestx(184)
worstxt(185)= bestx(185)
worstxt(186)= bestx(186)
worstxt(187)= bestx(187)
worstxt(188)= bestx(188)
worstxt(189)= bestx(189)
worstxt(190)= bestx(190)
worstxt(191)= bestx(191)
worstxt(192)= bestx(192)
worstxt(193)= bestx(193)
worstxt(194)= bestx(194)

```
            worstxt(195)= bestx(195)
            worstxt(196)= bestx(196)
            worstxt(197)= bestx(197)
            worstxt(198)= bestx(198)
            worstxt(199)= bestx(199)
            worstxt(200)= bestx(200)
            worstxt(201)= bestx(201)
            worstxt(202)= bestx(202)
            worstxt(203)= bestx(203)
            worstxt(204)= bestx(204)
            worstxt(205)= bestx(205)
            worstxt(206)= bestx(206)
            worstxt(207)= bestx(207)
            worstxt(208)= bestx(208)
            worstxt(209)= bestx(209)
            worstxt(210)= bestx(210)
          end if
          fvaltave = fvaltave + bestfval

          if ((elapsed(1)-ltime) .lt. besttt)
     +       besttt = elapsed(1)-ltime
          if ((elapsed(1)-ltime) .gt. worsttt)
     +       worsttt = elapsed(1)-ltime
          ttave = ttave + (elapsed(1)-ltime)

          if (real(iter) .lt. bestit)
     +       bestit = real(iter)
          if (real(iter) .gt. worstit)
     +       worstit = real(iter)
          itave = itave + real(iter)

          write (4, *) '];'

 9911 continue
          write (3, *) '];'
          write (5, *) '];'
          write (3, *) 'best = [', bestxt(1), bestxt(2), bestxt(3),
     + bestxt(4), bestxt(5), bestxt(6), bestxt(7), bestxt(8), bestxt(9),
     + bestxt(10),
     + bestxt(11), bestxt(12), bestxt(13), bestxt(14), bestxt(15),
     + bestxt(16), bestxt(17), bestxt(18), bestxt(19), bestxt(20),
     + bestxt(21), bestxt(22), bestxt(23), bestxt(24), bestxt(25),
     + bestxt(26), bestxt(27), bestxt(28), bestxt(29), bestxt(30),
     + bestxt(31), bestxt(32), bestxt(33), bestxt(34), bestxt(35),
     + bestxt(36), bestxt(37), bestxt(38), bestxt(39), bestxt(40),
     + bestxt(41), bestxt(42), bestxt(43), bestxt(44), bestxt(45),
     + bestxt(46), bestxt(47), bestxt(48), bestxt(49), bestxt(50),
     + bestxt(51), bestxt(52), bestxt(53), bestxt(54), bestxt(55),
     + bestxt(56), bestxt(57), bestxt(58), bestxt(59), bestxt(60),
     + bestxt(61), bestxt(62), bestxt(63), bestxt(64), bestxt(65),
     + bestxt(66), bestxt(67), bestxt(68), bestxt(69), bestxt(70),
     + bestxt(71), bestxt(72), bestxt(73), bestxt(74), bestxt(75),
     + bestxt(76), bestxt(77), bestxt(78), bestxt(79), bestxt(80),
     + bestxt(81), bestxt(82), bestxt(83), bestxt(84), bestxt(85),
     + bestxt(86), bestxt(87), bestxt(88), bestxt(89), bestxt(90),
     + bestxt(91), bestxt(92), bestxt(93), bestxt(94), bestxt(95),
     + bestxt(96), bestxt(97), bestxt(98), bestxt(99), bestxt(100),
     + bestxt(101), bestxt(102), bestxt(103), bestxt(104), bestxt(105),
     + bestxt(106), bestxt(107), bestxt(108), bestxt(109), bestxt(110),
     + bestxt(111), bestxt(112), bestxt(113), bestxt(114), bestxt(115),
     + bestxt(116), bestxt(117), bestxt(118), bestxt(119), bestxt(120),
     + bestxt(121), bestxt(122), bestxt(123), bestxt(124), bestxt(125),
     + bestxt(126), bestxt(127), bestxt(128), bestxt(129), bestxt(130),
     + bestxt(131), bestxt(132), bestxt(133), bestxt(134), bestxt(135),
     + bestxt(136), bestxt(137), bestxt(138), bestxt(139), bestxt(140),
     + bestxt(141), bestxt(142), bestxt(143), bestxt(144), bestxt(145),
     + bestxt(146), bestxt(147), bestxt(148), bestxt(149), bestxt(150),
     + bestxt(151), bestxt(152), bestxt(153), bestxt(154), bestxt(155),
     + bestxt(156), bestxt(157), bestxt(158), bestxt(159), bestxt(160),
     + bestxt(161), bestxt(162), bestxt(163), bestxt(164), bestxt(165),
     + bestxt(166), bestxt(167), bestxt(168), bestxt(169), bestxt(170),
     + bestxt(171), bestxt(172), bestxt(173), bestxt(174), bestxt(175),
     + bestxt(176), bestxt(177), bestxt(178), bestxt(179), bestxt(180),
     + bestxt(181), bestxt(182), bestxt(183), bestxt(184), bestxt(185),
     + bestxt(186), bestxt(187), bestxt(188), bestxt(189), bestxt(190),
     + bestxt(191), bestxt(192), bestxt(193), bestxt(194), bestxt(195),
     + bestxt(196), bestxt(197), bestxt(198), bestxt(199), bestxt(200),
     + bestxt(201), bestxt(202), bestxt(203), bestxt(204), bestxt(205),
     + bestxt(206), bestxt(207), bestxt(208), bestxt(209), bestxt(210),
     +       bestfvalt, bestit, besttt, '];'
```

```
          write (3, *) 'average = [', fvaltave/maxtrial,
     +                    itave/maxtrial, ttave/maxtrial, '];'
          write (3, *) 'worst = [', worstxt(1), worstxt(2), worstxt(3),
     + worstxt(4), worstxt(5), worstxt(6), worstxt(7), worstxt(8),
     + worstxt(9), worstxt(10),
     + worstxt(11), worstxt(12), worstxt(13), worstxt(14), worstxt(15),
     + worstxt(16), worstxt(17), worstxt(18), worstxt(19), worstxt(20),
     + worstxt(21), worstxt(22), worstxt(23), worstxt(24), worstxt(25),
     + worstxt(26), worstxt(27), worstxt(28), worstxt(29), worstxt(30),
     + worstxt(31), worstxt(32), worstxt(33), worstxt(34), worstxt(35),
     + worstxt(36), worstxt(37), worstxt(38), worstxt(39), worstxt(40),
     + worstxt(41), worstxt(42), worstxt(43), worstxt(44), worstxt(45),
     + worstxt(46), worstxt(47), worstxt(48), worstxt(49), worstxt(50),
     + worstxt(51), worstxt(52), worstxt(53), worstxt(54), worstxt(55),
     + worstxt(56), worstxt(57), worstxt(58), worstxt(59), worstxt(60),
     + worstxt(61), worstxt(62), worstxt(63), worstxt(64), worstxt(65),
     + worstxt(66), worstxt(67), worstxt(68), worstxt(69), worstxt(70),
     + worstxt(71), worstxt(72), worstxt(73), worstxt(74), worstxt(75),
     + worstxt(76), worstxt(77), worstxt(78), worstxt(79), worstxt(80),
     + worstxt(81), worstxt(82), worstxt(83), worstxt(84), worstxt(85),
     + worstxt(86), worstxt(87), worstxt(88), worstxt(89), worstxt(90),
     + worstxt(91), worstxt(92), worstxt(93), worstxt(94), worstxt(95),
     + worstxt(96), worstxt(97), worstxt(98), worstxt(99), worstxt(100),
     + worstxt(101), worstxt(102), worstxt(103), worstxt(104),
     + worstxt(105), worstxt(106), worstxt(107), worstxt(108),
     + worstxt(109), worstxt(110), worstxt(111), worstxt(112),
     + worstxt(113), worstxt(114), worstxt(115), worstxt(116),
     + worstxt(117), worstxt(118), worstxt(119), worstxt(120),
     + worstxt(121), worstxt(122), worstxt(123), worstxt(124),
     + worstxt(125), worstxt(126), worstxt(127), worstxt(128),
     + worstxt(129), worstxt(130), worstxt(131), worstxt(132),
     + worstxt(133), worstxt(134), worstxt(135), worstxt(136),
     + worstxt(137), worstxt(138), worstxt(139), worstxt(140),
     + worstxt(141), worstxt(142), worstxt(143), worstxt(144),
     + worstxt(145), worstxt(146), worstxt(147), worstxt(148),
     + worstxt(149), worstxt(150), worstxt(151), worstxt(152),
     + worstxt(153), worstxt(154), worstxt(155), worstxt(156),
     + worstxt(157), worstxt(158), worstxt(159), worstxt(160),
     + worstxt(161), worstxt(162), worstxt(163), worstxt(164),
     + worstxt(165), worstxt(166), worstxt(167), worstxt(168),
     + worstxt(169), worstxt(170), worstxt(171), worstxt(172),
     + worstxt(173), worstxt(174), worstxt(175), worstxt(176),
     + worstxt(177), worstxt(178), worstxt(179), worstxt(180),
     + worstxt(181), worstxt(182), worstxt(183), worstxt(184),
     + worstxt(185), worstxt(186), worstxt(187), worstxt(188),
     + worstxt(189), worstxt(190), worstxt(191), worstxt(192),
     + worstxt(193), worstxt(194), worstxt(195), worstxt(196),
     + worstxt(197), worstxt(198), worstxt(199), worstxt(200),
     + worstxt(201), worstxt(202), worstxt(203), worstxt(204),
     + worstxt(205), worstxt(206), worstxt(207), worstxt(208),
     + worstxt(209), worstxt(210),
     +                    worstfvalt, worstit, worsttt, '];'
          close(3)
          close(4)
          close(5)

          stop
          end

          function heatrate(no, megawt, mwlow, mwhi)

          implicit none
          integer no
          real heatrate, megawt, mwlow, mwhi, gcost, ocost
          real trfactor, opfactor
          real coef_a, coef_b, coef_c

*         Gas & Oil (Baht/GCal)
          gcost = 500.25
          ocost = 602.73

*         Transform BTU->Cal
          trfactor = 251.996

          if (megawt .lt. mwlow) then
            heatrate = 1000000000.0
            goto 9010
          else if (megawt .gt. mwhi) then
            heatrate = 1000000000.0
            goto 9010
```

```
      end if

      if (no .eq. 1) then
*     KN-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.010
      goto 9110
      else if (no .eq. 2) then
*     RPB-H1

      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.015
      goto 9110
      else if (no .eq. 3) then
*     RPB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 4) then
*     RPB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 5) then
*     MM-T8
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.010
      goto 9110
      else if (no .eq. 6) then
*     MM-T9
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.015
      goto 9110
      else if (no .eq. 7) then
*     MM-T4
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.000
      goto 9110
      else if (no .eq. 8) then
*     MM-T5
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.010
      goto 9110
      else if (no .eq. 9) then
*     MM-T6
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.015
      goto 9110
      else if (no .eq. 10) then
*     MM-T7
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.020
      goto 9110
      else if (no .eq. 11) then
*     SK-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.030
      goto 9110
      else if (no .eq. 12) then
*     SK-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.035
      goto 9110
      else if (no .eq. 13) then
*     SK-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.040
```

```
      goto 9110

      else if (no .eq. 14) then
*     BB-H8
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.045
      goto 9110
      else if (no .eq. 15) then
*     BB-H7
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.050
      goto 9110
      else if (no .eq. 16) then
*     BB-H6
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.055
      goto 9110
      else if (no .eq. 17) then
*     BB-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.060
      goto 9110
      else if (no .eq. 18) then
*     BB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.065
      goto 9110
      else if (no .eq. 19) then
*     NPO-C1
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
      else if (no .eq. 20) then
*     IPT-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.070
      goto 9110
      else if (no .eq. 21) then
*     BPK-C1
      coef_a = 22.4
      coef_b = 5760.0
      coef_c = 222000.0
      goto 9110
      else if (no .eq. 22) then
*     BPK-T2
      coef_a = 1.56
      coef_b = 8734.0
      coef_c = 59400.0
      goto 9110
      else if (no .eq. 23) then
*     SB-C1
      coef_a = -0.68
      coef_b = 6653.0
      coef_c = 114037.0
      goto 9110
      else if (no .eq. 24) then
*     SB-T1
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.0
      goto 9110
      else if (no .eq. 25) then
*     SB-T2
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.050
      goto 9110
      else if (no .eq. 26) then
*     SB-T3
      coef_a = 13.71
      coef_b = 6980.0
      coef_c = 141375.0
      goto 9110
```

```
      else if (no .eq. 27) then
*     KHL-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.075
      goto 9110
      else if (no .eq. 28) then
*     KHL-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.080
      goto 9110
      else if (no .eq. 29) then
*     KHL-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.085
      goto 9110
      else if (no .eq. 30) then
*     SNR-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.090
      goto 9110
      else if (no .eq. 31) then
*     SNR-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.095
      goto 9110
      else if (no .eq. 32) then
*     SNR-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.100
      goto 9110
      else if (no .eq. 33) then
*     SNR-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.105
      goto 9110
      else if (no .eq. 34) then
*     TN-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.110
      goto 9110
      else if (no .eq. 35) then
*     TN-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.115
      goto 9110
      else if (no .eq. 36) then
*     RB-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.120
      goto 9110
      else if (no .eq. 37) then
*     RB-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.125
      goto 9110
      else if (no .eq. 38) then
*     THB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.130
      goto 9110
      else if (no .eq. 39) then
*     THB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.135
      goto 9110
      else if (no .eq. 40) then
*     GCC-T1
```

```
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.140
      goto 9110
      else if (no .eq. 41) then
*     HH-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.145
      goto 9110
      else if (no .eq. 42) then
*     HH-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.150
      goto 9110
      else if (no .eq. 43) then
*     AEP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.155
      goto 9110
      else if (no .eq. 44) then
*     TPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.160
      goto 9110
      else if (no .eq. 45) then
*     TTP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.165
      goto 9110
      else if (no .eq. 46) then
*     NPC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.170
      goto 9110
      else if (no .eq. 47) then

*     IP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.175
      goto 9110
      else if (no .eq. 48) then
*     IP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.180
      goto 9110
      else if (no .eq. 49) then
*     BCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.185
      goto 9110
      else if (no .eq. 50) then
*     MTP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.190
      goto 9110
      else if (no .eq. 51) then
*     MTP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.195
      goto 9110
      else if (no .eq. 52) then
*     TCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.200
      goto 9110
      else if (no .eq. 53) then
*     TCC-T2
      coef_a = 0.0
```

```
      coef_b = 0.01
      coef_c = 0.205
      goto 9110
      else if (no .eq. 54) then
*     C0C0-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.210
      goto 9110
      else if (no .eq. 55) then
*     C0C0-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.215
      goto 9110
      else if (no .eq. 56) then
*     NPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 57) then
*     NPS-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 58) then
*     NPO-C2
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
      else if (no .eq. 59) then
*     WN-C3
      coef_a = -5.55
      coef_b = 7455.0
      coef_c = 50631.0
      goto 9110
      else if (no .eq. 60) then
*     KN-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.230
      goto 9110
      else if (no .eq. 61) then
*     RY-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.235
      goto 9110
      else if (no .eq. 62) then
*     RY-C4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.240
      goto 9110
      else if (no .eq. 63) then
*     BB-H5
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.245
      goto 9110
      else if (no .eq. 64) then
*     BB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.250
      goto 9110
      else if (no .eq. 65) then
*     BB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.255
      goto 9110
      else if (no .eq. 66) then
*     BLCP-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.260
```

```
      goto 9110
      else if (no .eq. 67) then
*     BLCP-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.265
      goto 9110
      else if (no .eq. 68) then
*     EPEC-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.270
      goto 9110
      else if (no .eq. 69) then
*     GULF-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.275
      goto 9110
      else if (no .eq. 70) then
*     GULF-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.280
      goto 9110
      end if

 9110   opfactor = megawt/mwhi*100.0
       heatrate = gcost * trfactor *
     +    (coef_a*opfactor**2 + coef_b*opfactor + coef_c)
     +    / 1000000.0

 9010 return
      end

      subroutine chkmw(megawt, mwlow, mwhi)
      implicit none
      real megawt, mwlow, mwhi
      if (megawt .lt. mwlow) then
        megawt = mwlow
      else if (megawt .gt. mwhi) then
        megawt = mwhi
      end if
      return
      end

      subroutine ran3(idum,rand)

      implicit double precision (a-h,m,o-z)
      save
*     implicit real*4(m)
      parameter (mbig=4000000.,mseed=1618033.,mz=0.,fac=1./mbig)
*     parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1./mbig)
*
*     According to Knuth, any large mbig, and any smaller (but still large)
*     mseed can be substituted for the above values.
*
      dimension ma(55)
      data iff /0/
      if (idum.lt.0 .or. iff.eq.0) then
        iff=1
        mj=mseed-dble(iabs(idum))
        mj=dmod(mj,mbig)
        ma(55)=mj
        mk=1
        do 9010 i=1,54
           ii=mod(21*i,55)
           ma(ii)=mk
           mk=mj-mk
           if(mk.lt.mz) mk=mk+mbig
           mj=ma(ii)
 9010    continue
        do 9015 k=1,4
           do 9020 i=1,55
              ma(i)=ma(i)-ma(1+mod(i+30,55))
              if(ma(i).lt.mz) ma(i)=ma(i)+mbig
 9020       continue
 9015    continue
        inext=0
        inextp=31
```

```
        idum=1
      endif
      inext=inext+1
      if(inext.eq.56) inext=1
      inextp=inextp+1
      if(inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
      rand=mj*fac
      return
      end
```

ค.3 วิธีนิชชิ่งเจเนติกอัลกอริทึมร่วมกับตัวปฏิบัติการทางพันธุศาสตร์ที่มีประสิทธิภาพสูง

```
      program dedaonga
*
*     This program performs economic dispatch algorithms
*     for economic generation dispatch problems.
*
      implicit none

      integer maxgen, maxiter, iter, szpop, szchr, szind, szchild
      integer pop, popc, bestind, stamax, maxtrial, trapped, trapmax
      integer idum, ichr, ichild, i, j, itry, ic, iv
      integer m, n, ii, jj, iii, hrfn, iperiod, itmp, damaxgen
      integer mate, sztour, ipick, station, intval, chrtmp
      real k1, k2, k3, k4, rtmp, c
      real bestfit, bestfval, bestx, fitness, fitnessn
      real poppow, popcx, etime, elapsed(2), ltime, total
      real pcross, pmutate, fitmax, fitave, fitdev, pmumax
      real bestfvalt, worstfvalt, bestxt, worstxt, fvaltave
      real besttt, worsttt, ttave
      real bestit, worstit, itave
      real losstmp, tmp, bias, biasac
      real bcoeff, b0, b00, sumpow, heatrate
      real sumcost, costtmp, sumpowtmp, mustrunmw
      real mwnow, summw, sysloss, lowlim, hilim, mwmin, mwmax
      real rampdn, rampup, osysloss, osummw, edemand, odemand
      real coef_b, coef_c, tempb, Pr1, Pr2, remain
      real opmax, opmin, pmax, pmin, loadf
      real dp1c1, dp2c2, dp1c2, dp2c1, fitc1, fitc2
      logical forceup, forcedn, debug
      double precision val
      double precision fval, fvaltmp

      parameter (maxgen = 70, damaxgen = 15)
      parameter (szchild = 2, sztour = 4, intval = 3)
      parameter (szchr = 10, szind = (damaxgen-1)*szchr*intval)
      parameter (szpop = szind*2)

      dimension mate(szchild)
      dimension pop(szpop, szind), popc(szpop, szind)
      dimension poppow(szpop, maxgen*intval)
      dimension popcx(szpop, maxgen*intval)
      dimension fitness(szpop), fitnessn(szpop)
      dimension bestind(szind), bestx(maxgen*intval)
      dimension bestxt(maxgen*intval), worstxt(maxgen*intval)
      dimension mwnow(maxgen), lowlim(maxgen), hilim(maxgen)
      dimension mwmin(maxgen), mwmax(maxgen)
      dimension hrfn(maxgen), losstmp(maxgen), tempb(maxgen-1)
      dimension bcoeff(maxgen, maxgen), b0(maxgen)
      dimension loadf(intval), chrtmp(szchr)
      dimension rampdn(maxgen), rampup(maxgen)

*      data mwnow/maxgen*400.0/
*      data rampup/maxgen*50.0/
*      data rampdn/maxgen*80.0/

       mwnow(1) = 70.0
       mwnow(2) = 79.0
       mwnow(3) = 79.0
       mwnow(4) = 79.0
       mwnow(5) = 150.0
       mwnow(6) = 150.0
```

```
 mwnow(7) = 90.0
 mwnow(8) = 90.0
 mwnow(9) = 90.0
 mwnow(10) = 90.0
 mwnow(11) = 112.0
 mwnow(12) = 111.0
 mwnow(13) = 102.0
 mwnow(14) = 152.0
 mwnow(15) = 115.0
 mwnow(16) = 76.0
mwnow(17) = 76.0
 mwnow(18) = 76.0
 mwnow(19) = 180.0
 mwnow(20) = 700.0
 mwnow(21) = 320.0
 mwnow(22) = 693.0
 mwnow(23) = 284.0
 mwnow(24) = 140.0
 mwnow(25) = 140.0
 mwnow(26) = 140.0
 mwnow(27) = 88.0
 mwnow(28) = 88.0
 mwnow(29) = 88.0
 mwnow(30) = 120.0
 mwnow(31) = 120.0
 mwnow(32) = 120.0
 mwnow(33) = 180.0
 mwnow(34) = 20.0
 mwnow(35) = 20.0
 mwnow(36) = 700.0
 mwnow(37) = 700.0
 mwnow(38) = 105.0
 mwnow(39) = 105.0
 mwnow(40) = 90.0
 mwnow(41) = 63.0
 mwnow(42) = 63.0
 mwnow(43) = 90.0
 mwnow(44) = 25.0
 mwnow(45) = 10.0
 mwnow(46) = 32.0
 mwnow(47) = 55.0
 mwnow(48) = 55.0
 mwnow(49) = 90.0
 mwnow(50) = 60.0
 mwnow(51) = 60.0
 mwnow(52) = 90.0
 mwnow(53) = 90.0
 mwnow(54) = 90.0
 mwnow(55) = 90.0
 mwnow(56) = 90.0
 mwnow(57) = 90.0
 mwnow(58) = 180.0
 mwnow(59) = 250.0
 mwnow(60) = 70.0
 mwnow(61) = 292.0
 mwnow(62) = 292.0
 mwnow(63) = 76.0
 mwnow(64) = 76.0
 mwnow(65) = 76.0
 mwnow(66) = 673.0
 mwnow(67) = 673.0
 mwnow(68) = 350.0
 mwnow(69) = 350.0
 mwnow(70) = 350.0

 rampup(1) = 5.0*6.8
 rampup(2) = 5.0*40.0
 rampup(3) = 5.0*40.0
 rampup(4) = 5.0*40.0
 rampup(5) = 5.0*5.0
 rampup(6) = 5.0*5.0
 rampup(7) = 5.0*3.0
 rampup(8) = 5.0*3.0
 rampup(9) = 5.0*3.0
 rampup(10) = 5.0*3.0
 rampup(11) = 5.0*70.0
 rampup(12) = 5.0*70.0
 rampup(13) = 5.0*70.0
 rampup(14) = 5.0*120.0
 rampup(15) = 5.0*115.0
```

```
rampup(16) = 5.0*76.3
rampup(17) = 5.0*76.3
rampup(18) = 5.0*76.3
rampup(19) = 5.0*8.0
rampup(20) = 5.0*39.0
rampup(21) = 5.0*22.5
rampup(22) = 5.0*25.0
rampup(23) = 5.0*10.0
rampup(24) = 5.0*35.0
rampup(25) = 5.0*35.0
rampup(26) = 5.0*35.0
rampup(27) = 5.0*87.0
rampup(28) = 5.0*87.0
rampup(29) = 5.0*87.0
rampup(30) = 5.0*70.2
rampup(31) = 5.0*70.2
rampup(32) = 5.0*70.2
rampup(33) = 5.0*70.2
rampup(34) = 5.0*20.0
rampup(35) = 5.0*20.0
rampup(36) = 5.0*39.0
rampup(37) = 5.0*39.0
rampup(38) = 5.0*70.0
rampup(39) = 5.0*70.0
rampup(40) = 5.0*35.0
rampup(41) = 5.0*63.0
rampup(42) = 5.0*63.0
rampup(43) = 5.0*35.0
rampup(44) = 5.0*35.0
rampup(45) = 5.0*10.0
rampup(46) = 5.0*35.0
rampup(47) = 5.0*35.0
rampup(48) = 5.0*35.0
rampup(49) = 5.0*35.0
rampup(50) = 5.0*35.0
rampup(51) = 5.0*35.0
rampup(52) = 5.0*35.0
rampup(53) = 5.0*35.0
rampup(54) = 5.0*35.0
rampup(55) = 5.0*35.0
rampup(56) = 5.0*35.0
rampup(57) = 5.0*35.0

rampup(58) = 5.0*8.0
rampup(59) = 5.0*39.0
rampup(60) = 5.0*6.8
rampup(61) = 5.0*55.0
rampup(62) = 5.0*55.0
rampup(63) = 5.0*76.3
rampup(64) = 5.0*76.3
rampup(65) = 5.0*76.3
rampup(66) = 5.0*10.0
rampup(67) = 5.0*10.0
rampup(68) = 5.0*8.0
rampup(69) = 5.0*8.0
rampup(70) = 5.0*8.0

rampdn(1) = 5.0*6.8
rampdn(2) = 5.0*40.0
rampdn(3) = 5.0*40.0
rampdn(4) = 5.0*40.0
rampdn(5) = 5.0*5.0
rampdn(6) = 5.0*5.0
rampdn(7) = 5.0*3.0
rampdn(8) = 5.0*3.0
rampdn(9) = 5.0*3.0
rampdn(10) = 5.0*3.0
rampdn(11) = 5.0*70.0
rampdn(12) = 5.0*70.0
rampdn(13) = 5.0*70.0
rampdn(14) = 5.0*120.0
rampdn(15) = 5.0*115.0
rampdn(16) = 5.0*76.3
rampdn(17) = 5.0*76.3
rampdn(18) = 5.0*76.3
rampdn(19) = 5.0*8.0
rampdn(20) = 5.0*39.0
rampdn(21) = 5.0*22.5
rampdn(22) = 5.0*25.0
rampdn(23) = 5.0*10.0
```

```
        rampdn(24) = 5.0*35.0
        rampdn(25) = 5.0*35.0
        rampdn(26) = 5.0*35.0
        rampdn(27) = 5.0*87.0
        rampdn(28) = 5.0*87.0
        rampdn(29) = 5.0*87.0
        rampdn(30) = 5.0*70.2
        rampdn(31) = 5.0*70.2
        rampdn(32) = 5.0*70.2
        rampdn(33) = 5.0*70.2
        rampdn(34) = 5.0*20.0
        rampdn(35) = 5.0*20.0
        rampdn(36) = 5.0*39.0
        rampdn(37) = 5.0*39.0
        rampdn(38) = 5.0*70.0
        rampdn(39) = 5.0*70.0
        rampdn(40) = 5.0*35.0
        rampdn(41) = 5.0*63.0
        rampdn(42) = 5.0*63.0
        rampdn(43) = 5.0*35.0
        rampdn(44) = 5.0*35.0
        rampdn(45) = 5.0*10.0
        rampdn(46) = 5.0*35.0
        rampdn(47) = 5.0*35.0
        rampdn(48) = 5.0*35.0
        rampdn(49) = 5.0*35.0
        rampdn(50) = 5.0*35.0
        rampdn(51) = 5.0*35.0
        rampdn(52) = 5.0*35.0
        rampdn(53) = 5.0*35.0
        rampdn(54) = 5.0*35.0
        rampdn(55) = 5.0*35.0
        rampdn(56) = 5.0*35.0
        rampdn(57) = 5.0*35.0
        rampdn(58) = 5.0*8.0
        rampdn(59) = 5.0*39.0
        rampdn(60) = 5.0*6.8
        rampdn(61) = 5.0*55.0
        rampdn(62) = 5.0*55.0
        rampdn(63) = 5.0*76.3
        rampdn(64) = 5.0*76.3
        rampdn(65) = 5.0*76.3
        rampdn(66) = 5.0*10.0
        rampdn(67) = 5.0*10.0
        rampdn(68) = 5.0*8.0
        rampdn(69) = 5.0*8.0
        rampdn(70) = 5.0*8.0

      loadf(1) = 1.002
      loadf(2) = 1.005
      loadf(3) = 1.07
*      loadf(4) = 0.985
*      loadf(5) = 0.992
*      loadf(6) = 1.015

      open(unit = 1, file = 'b.dat', status = 'old')

      do 1 i = 1, maxgen
         do 2 j = 1, maxgen
            read (1, *) bcoeff(i, j)
    2    continue
    1 continue
      close(1)

      open(unit = 2, file = 'b0.dat', status = 'old')
      do 3 i = 1, maxgen
         read (2, *) b0(i)
    3 continue
      read(2,*) b00
      close(2)

      lowlim(1) = 60.0
      lowlim(2) = 50.0
      lowlim(3) = 50.0
      lowlim(4) = 50.0
      lowlim(5) = 150.0
      lowlim(6) = 150.0
      lowlim(7) = 90.0
      lowlim(8) = 90.0
      lowlim(9) = 90.0
```

```
lowlim(10) = 90.0
lowlim(11) = 60.0
lowlim(12) = 60.0
lowlim(13) = 60.0
lowlim(14) = 60.0
lowlim(15) = 60.0
lowlim(16) = 40.0
lowlim(17) = 40.0
lowlim(18) = 40.0
lowlim(19) = 180.0
lowlim(20) = 350.0
lowlim(21) = 315.0
lowlim(22) = 280.0
lowlim(23) = 284.0
lowlim(24) = 140.0
lowlim(25) = 140.0
lowlim(26) = 140.0
lowlim(27) = 50.0
lowlim(28) = 50.0
lowlim(29) = 50.0
lowlim(30) = 70.0
lowlim(31) = 70.0
lowlim(32) = 70.0
lowlim(33) = 120.0
lowlim(34) = 15.0
lowlim(35) = 15.0
lowlim(36) = 250.0
lowlim(37) = 250.0
lowlim(38) = 20.0
lowlim(39) = 20.0
lowlim(40) = 59.0
lowlim(41) = 10.0
lowlim(42) = 10.0
lowlim(43) = 59.0
lowlim(44) = 17.0
lowlim(45) = 5.0
lowlim(46) = 21.0
lowlim(47) = 36.0
lowlim(48) = 36.0
lowlim(49) = 59.0
lowlim(50) = 39.0
lowlim(51) = 39.0
lowlim(52) = 59.0
lowlim(53) = 59.0
lowlim(54) = 59.0
lowlim(55) = 59.0
lowlim(56) = 59.0
lowlim(57) = 59.0
lowlim(58) = 180.0
lowlim(59) = 250.0
lowlim(60) = 60.0
lowlim(61) = 250.0
lowlim(62) = 250.0
lowlim(63) = 40.0
lowlim(64) = 40.0
lowlim(65) = 40.0
lowlim(66) = 200.0
lowlim(67) = 200.0
lowlim(68) = 100.0
lowlim(69) = 100.0
lowlim(70) = 100.0

hilim(1) = 70.0
hilim(2) = 79.0
hilim(3) = 79.0
hilim(4) = 79.0
hilim(5) = 276.0
hilim(6) = 276.0
hilim(7) = 140.0
hilim(8) = 140.0
hilim(9) = 140.0
hilim(10) = 140.0
hilim(11) = 112.0
hilim(12) = 111.0
hilim(13) = 102.0
hilim(14) = 152.0
hilim(15) = 115.0
hilim(16) = 76.0
hilim(17) = 76.0
hilim(18) = 76.0
```

```
      hilim(19) = 330.0
      hilim(20) = 700.0
      hilim(21) = 360.0
      hilim(22) = 700.0
      hilim(23) = 316.0

      hilim(24) = 735.0
      hilim(25) = 735.0
      hilim(26) = 735.0
      hilim(27) = 88.0
      hilim(28) = 88.0
      hilim(29) = 88.0
      hilim(30) = 120.0
      hilim(31) = 120.0
      hilim(32) = 120.0
      hilim(33) = 180.0
      hilim(34) = 20.0
      hilim(35) = 20.0
      hilim(36) = 700.0
      hilim(37) = 700.0
      hilim(38) = 105.0
      hilim(39) = 105.0
      hilim(40) = 90.0
      hilim(41) = 63.0
      hilim(42) = 63.0
      hilim(43) = 90.0
      hilim(44) = 25.0
      hilim(45) = 10.0
      hilim(46) = 32.0
      hilim(47) = 55.0
      hilim(48) = 55.0
      hilim(49) = 90.0
      hilim(50) = 60.0
      hilim(51) = 60.0
      hilim(52) = 90.0
      hilim(53) = 90.0
      hilim(54) = 90.0
      hilim(55) = 90.0
      hilim(56) = 90.0
      hilim(57) = 90.0
      hilim(58) = 330.0
      hilim(59) = 700.0
      hilim(60) = 70.0
      hilim(61) = 292.0
      hilim(62) = 292.0
      hilim(63) = 76.0
      hilim(64) = 76.0
      hilim(65) = 76.0
      hilim(66) = 673.0
      hilim(67) = 673.0
      hilim(68) = 350.0
      hilim(69) = 350.0
      hilim(70) = 350.0

      do 4 i = 1, maxgen
         hrfn(i) = i
         tmp = mwnow(i) - rampdn(i)
         call chkmw(tmp, lowlim(i), hilim(i))
         mwmin(i) = tmp
         tmp = mwnow(i) + rampup(i)
         call chkmw(tmp, lowlim(i), hilim(i))
         mwmax(i) = tmp
*        print *,
*     + mwmin(i), mwnow(i), mwmax(i)

    4 continue
*
*     Determines the system loss.
*         System loss = PiBijPj + Bi0Pi + B00
*
            sysloss = 0.0
            summw = 0.0
            sumcost = 0.0
            do 6140 m = 1, maxgen
               losstmp(m) = 0.0
               summw = summw + mwnow(m)
               sumcost = sumcost + heatrate(hrfn(m),
     + mwnow(m), lowlim(m), hilim(m))
               do 6145 n = 1, maxgen
                  losstmp(m) = losstmp(m) +
```

```
     +                   mwnow(n)*bcoeff(n,m)/100.0
 6145          continue
               sysloss = sysloss + losstmp(m)*mwnow(m)/100.0
 6140       continue

            do 6150 m = 1, maxgen
               sysloss = sysloss + mwnow(m)*b0(m)/100.0
 6150       continue
            sysloss = (sysloss + b00)*100.0
            odemand = summw - sysloss
            osysloss = sysloss
            osummw = summw
            print *, 'System test :  '
            print *, 'odemand = ', odemand,
     +      'osummw = ', osummw, 'osysloss (%) = ',
     +      osysloss/osummw*100.0
*           pause
*
*     Normalizes participation values
*         partcp = 0.xxxxx
*                  |---|
*                 nchrlen
*
      edemand = loadf(1)*odemand
      print *, 'Demand = ', edemand
      maxtrial = 100
      stamax = 100
      maxiter = 1000
      k1 = 1.0
      k2 = 0.5
      k3 = 1.0
      k4 = 0.5
      idum = -14000
      pmumax = 0.15
      biasac = 2.0
      trapmax = 4

*     ********************************************************
*                    Program Start
*     ********************************************************
      open(unit = 3, file = 'ongad.m', status = 'new')
      open(unit = 4, file = 'fitngad.m', status = 'new')
      open(unit = 5, file = 'ctngad.m', status = 'new')

      bestfvalt = 100000000000.0
      worstfvalt = 0.0
      fvaltave = 0.0
      besttt = 100000000.0
      worsttt = 0.00000001
      ttave = 0.0
      bestit = 100000000.0
      worstit = 1.0
      itave = 0.0
      write (3, *) 'oga = ['
      write (5, *) 'ct = ['

      do 9911 itry = 1, maxtrial
      print *, '###### Trial : ', itry, ' #######'
      write (4, *) 'fit', itry, ' = ['

      ic = 0
      iv = 0
      total = etime(elapsed)
      ltime = elapsed(1)
      bestfit = 0.0
      ipick = szpop
      station = 0
*     Initial random number generator
      call ran3(idum, val)
      do 5 i = 1, szpop
         do 10 j = 1, szind
            pop(i, j) = 1
            call ran3(1, val)
            if (val .lt. 0.5) pop(i, j) = 0
   10    continue
    5 continue

      do 920 iter = 1, maxiter
         print *, '###### Generation : ', iter, ' #######'
         fitave = 0.0
```

```
         fitdev = 0.0
         fitmax = 0.0

         do 25 ichr = 1, szpop
*             print *, '###### Individual : ', ichr, ' #######'
              fval = 0.0
         do 1122 iperiod = 1, intval
*             print *, '###### Period : ', iperiod, ' #######'
              bias = 0.0
             trapped = 0
             if (iperiod .eq. 1) then
              do 8011 i = 1, maxgen
                 tmp = mwnow(i) - rampdn(i)
                 call chkmw(tmp, lowlim(i), hilim(i))
                 mwmin(i) = tmp
                 tmp = mwnow(i) + rampup(i)
                 call chkmw(tmp, lowlim(i), hilim(i))
                 mwmax(i) = tmp
*                print *,
*     + mwmin(i), mwmax(i)
 8011         continue
             else
              do 8022 i = 1, maxgen
                 tmp = poppow(ichr, ((iperiod-2)*(maxgen) + i))
     +           - rampdn(i)
                 call chkmw(tmp, lowlim(i), hilim(i))
                 mwmin(i) = tmp
                 tmp = poppow(ichr, ((iperiod-2)*(maxgen) + i))
     +           + rampup(i)
                 call chkmw(tmp, lowlim(i), hilim(i))
                 mwmax(i) = tmp
*                print *,
*     + mwmin(i), poppow(ichr, ((iperiod-2)*(maxgen) + i)), mwmax(i)
 8022         continue
             end if

                opmax = 0.0
                opmin = 0.0
                mustrunmw = 0.0
                do 3331 j = 1, maxgen-1
                   if ((j .ge. 1) .and. (j .le. 4)) then
                      mustrunmw = mustrunmw + mwmax(j)
                      goto 3331
                    end if
                     if ((j .ge. 11) .and. (j .le. 18)) then
                      mustrunmw = mustrunmw + mwmax(j)
                      goto 3331
                    end if
                     if (j .eq. 20) then
                      mustrunmw = mustrunmw + mwmax(j)
                       goto 3331
                    end if
                     if ((j .ge. 27) .and. (j .le. 57)) then
                      mustrunmw = mustrunmw + mwmax(j)
                       goto 3331
                    end if
                     if ((j .ge. 60) .and. (j .le. 70)) then
                      mustrunmw = mustrunmw + mwmax(j)
                       goto 3331
                    end if

                     opmax = opmax + mwmax(j)
                     opmin = opmin + mwmin(j)
 3331           continue

 2882        fvaltmp = 0.0
*            Check edemand
             pmax = opmax
            pmin = opmin
             remain = loadf(iperiod)*odemand+osysloss*loadf(iperiod)
     + -  (mwmax(maxgen)+mwmin(maxgen))/2.0 - mustrunmw
             sumpowtmp = 0.0
            forceup = .false.
            forcedn = .false.

            ii = 0
            do 30 i = 1, maxgen-1
*               Check dispatchable gernerating units
                if ((i .ge. 1) .and. (i .le. 4)) then
                  poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
```

```
                goto 3333
              end if

              if ((i .ge. 11) .and. (i .le. 18)) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                goto 3333
              end if

              if (i .eq. 20) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                goto 3333
              end if

              if ((i .ge. 27) .and. (i .le. 57)) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                goto 3333
              end if

              if ((i .ge. 60) .and. (i .le. 70)) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                goto 3333
              end if

              ii = ii + 1
              if (forceup) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                do 653 m = 1, szchr
*                   Engraying...
                  if (m .eq. 1) then
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = 1
                  else
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = 0
                  end if
 653            continue
          remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)

                goto 3333
              end if

              if (forcedn) then
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
                do 654 m = 1, szchr
*                   Engraying...
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +      = 0
 654            continue
          remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)

                goto 3333
              end if

          if (bias .ne. 0.0) then
       tmp = poppow(ichr, ((iperiod-1)*(maxgen) + i)) + bias

       if (tmp .gt. mwmax(i)) then
         bias = bias - mwmax(i) +
     +                 poppow(ichr, ((iperiod-1)*(maxgen) + i))
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
       else if (tmp .lt. mwmin(i)) then
         bias = bias - mwmin(i) +
     +                 poppow(ichr, ((iperiod-1)*(maxgen) + i))
                poppow(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
       else
           poppow(ichr, ((iperiod-1)*(maxgen) + i)) = tmp
           bias = 0.0
       end if
*                  Re-encoding...
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
              do 2991 m = 1, szchr
                 if (tmp .gt. 2.0**-m) then
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +   (szchr*(ii-1)+szchr+1-m))) = 1
                   tmp = tmp - 2.0**-m
```

```
                  else
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +    (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 2991             continue
*                 Re-engraying...
                  do 2992 m = 2, szchr
       itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 2992             continue
                  do 2993 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 2993             continue
          else
               poppow(ichr, ((iperiod-1)*(maxgen) + i)) = 0.0
               itmp = 0
*               Degraying
               do 35 j = 1, szchr
          itmp = itmp + pop(ichr,
     + ((iperiod-1)*(damaxgen-1)*szchr+(szchr*(ii-1)+j)))
                  itmp = itmp-(itmp/2)*2
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + poppow(ichr, ((iperiod-1)*(maxgen) + i)) +
     + real(itmp)*2.0**-(szchr-j+1)
   35          continue
                 poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + mwmin(i) + poppow(ichr, ((iperiod-1)*(maxgen) + i))*
     + (mwmax(i) - mwmin(i))
            end if
            remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
               pmax = pmax - mwmax(i)
               pmin = pmin - mwmin(i)
*
*    ------------------------------------------------------
*
            if (ii .lt. damaxgen-1) then
               if (remain .ge. pmax) then
                  forceup = .true.
       remain = remain + poppow(ichr, ((iperiod-1)*(maxgen) + i))
       poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmax
*       if ((poppow(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (poppow(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     +  then
*        print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
*       print *, 'remain .ge. pmax ->', remain
       remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
*                 Re-encoding...
*       print *, 'remain (new) ->', remain
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                  do 2901 m = 1, szchr
                     if (tmp .gt. 2.0**-m) then
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                 (szchr*(ii-1)+szchr+1-m))) = 1
                     tmp = tmp - 2.0**-m
                  else
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                 (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 2901             continue
*                 Re-engraying...
                  do 2902 m = 2, szchr
       itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 2902             continue
                  do 2903 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 2903             continue
                end if
```

```
               if (remain .le. pmin) then
                   forcedn = .true.
       remain = remain + poppow(ichr, ((iperiod-1)*(maxgen) + i))
*       print *, 'In first:', poppow(ichr,((iperiod-1)*(maxgen) + i))
*     + , mwmax(i), mwmin(i), remain
                  poppow(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmin
*       if ((poppow(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (poppow(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     + then
*        print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
          remain = remain - poppow(ichr, ((iperiod-1)*(maxgen) + i))
*                 Re-encoding...
       tmp = (poppow(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
*       print *, 'check ', poppow(ichr, ((iperiod-1)*(maxgen) + i)),
*     +              mwmin(i), remain
                  do 2904 m = 1, szchr
                     if (tmp .gt. 2.0**-m) then
         pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                       (szchr*(ii-1)+szchr+1-m))) = 1
                       tmp = tmp - 2.0**-m
                     else
         pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                       (szchr*(ii-1)+szchr+1-m))) = 0
                     end if
 2904             continue
*                 Re-engraying...
                  do 2905 m = 2, szchr
        itmp = pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) +
     +   pop(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +      (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 2905             continue
                  do 2906 m = 2, szchr
       pop(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     + = chrtmp(m)
 2906             continue
               end if
           end if

*
*              Check power participation
*

 3333       sumpowtmp = sumpowtmp + poppow(ichr,
     + ((iperiod-1)*(maxgen) + i))

*               print *, 'Ind index = ', ichr, ' Gen = ', iter,
*     + 'Period = ', iperiod
*               print *, 'Pgen', i, ' = ', poppow(ichr,
*     + ((iperiod-1)*(maxgen) + i)), ' remain = ', remain, ' pmax = ',
*     + pmax, ' pmin = ', pmin
*               print *, 'forceup = ', forceup, ' forcedn = ', forcedn
*               print *, 'sumpowtmp = ', sumpowtmp
*               pause

              costtmp = heatrate(hrfn(i),
     + poppow(ichr, ((iperiod-1)*(maxgen) + i)), lowlim(i),
     + hilim(i))
*              fval (MBaht/h)
              fvaltmp = fvaltmp + costtmp/100000000.0
*             print *, 'Total cost =', fvaltmp
   30       continue
*
*     Determine Pr
*          Demand - SumMW + Ploss - Pr = 0
*          Ploss = aPr^2 + bPr + c
*              a = Brr
*              b = BraPa + PatBar + B0r
*              c = PatBaaPa + PatB0a + B00
*          Thus, modify coefficients.
*              a = Brr
*              b = b-1 = BraPa + PatBar + B0r - 1
*              c = PatBaaPa + PatB0a + B00 + Demand - sumMW
```

```
*          Determines Pr
*             Pr = -b +- sqrt(b^2 - 4*a*c)
*                  ----------------------
*                            2*a
           coef_b = 0.0
           coef_c = 0.0

           do 8903 ii = 1, maxgen-1
              coef_b = coef_b +
     + bcoeff(maxgen, ii)*poppow(ichr, ((iperiod-1)*(maxgen-1) +
     + ii))/100.0
              tempb(ii) = 0.0
 8903      continue
           coef_b = coef_b*2.0 + b0(maxgen) - 1.0

           do 8904 ii = 1, maxgen-1
              do 8905 iii = 1, maxgen-1
                 tempb(ii) = tempb(ii) +
     + poppow(ichr, ((iperiod-1)*(maxgen)+iii))*bcoeff(iii, ii)/100.0
 8905         continue
 8904      continue
           do 8906 ii = 1, maxgen-1
              coef_c = coef_c + tempb(ii)*
     + poppow(ichr, ((iperiod-1)*(maxgen) + ii))/100.0
 8906      continue
           do 8907 ii = 1, maxgen-1
              coef_c = coef_c + poppow(ichr,((iperiod-1)*(maxgen)+ii))
     +        *b0(ii)/100.0
 8907      continue
       coef_c = coef_c + b00 + (loadf(iperiod)*odemand-sumpowtmp)
     +    /100.0
*           print *, 'load', loadf(iperiod)*odemand, sumpowtmp
           Pr1 = -1.0*coef_b + sqrt(coef_b*coef_b -
     +                         4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr1 = 100.0*Pr1/(2.0*bcoeff(maxgen,maxgen))
           Pr2 = -1.0*coef_b - sqrt(coef_b*coef_b -
     +                         4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr2 = 100.0*Pr2/(2.0*bcoeff(maxgen,maxgen))

           if (Pr1 .le. 0.0) then
              if (Pr2 .gt. 0.0) Pr1 = Pr2
           end if
           if (Pr1 .gt. Pr2) Pr1 = Pr2
           sumpowtmp = sumpowtmp + Pr1
           poppow(ichr, (iperiod-1)*maxgen + maxgen) = Pr1

           if (Pr1 .gt. mwmax(maxgen)) then
*              print *, 'Power imbalance occurred. (Pr1 > mwmax)',
*     +            Pr1, '>', mwmax(maxgen),
*     +      '(IChr = ', ichr, ' Period = ', iperiod, ')'
            if (trapped .gt. trapmax) then
                 print *, 'Trapped'
*             Re-building the chromosome
              do 7090 ii = 1, szchr*(damaxgen-1)
                 pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +            + ii)) = 1
                 call ran3(1, val)
                 if (val .lt. 0.5) then
                    pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +            + ii)) = 0
                  end if
 7090         continue
              trapped = 0
              goto 2882
            end if
              bias = Pr1 - mwmax(maxgen)
*             Loss compensating...
              bias = bias + bias*sysloss/sumpowtmp*biasac
*             print *, 'bias = ', bias, sysloss
              trapped = trapped + 1
              goto 2882
           end if
           if (Pr1 .lt. mwmin(maxgen)) then
*              print *, 'Power imbalance occurred. (Pr1 < mwmin)',
*     +      Pr1, '<', mwmin(maxgen),
*     +      '(IChr = ', ichr, ' Period = ', iperiod, ')'
            if (trapped .gt. trapmax) then
                 print *, 'Trapped'
*             Re-building the chromosome
              do 7091 ii = 1, szchr*(damaxgen-1)
```

```
                pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   pop(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 0
                 end if
7091          continue
              trapped = 0
              goto 2882
             end if
               bias = Pr1 - mwmin(maxgen)
*             Loss compensating...
              bias = bias + bias*sysloss/sumpowtmp*biasac
*             print *, 'bias = ', bias, sysloss
              trapped = trapped + 1
*             print *, trapped
              goto 2882
            end if
            costtmp = heatrate(hrfn(maxgen), Pr1, mwmin(maxgen),
     + mwmax(maxgen))
           sumpow = sumpowtmp
*          fval (MBaht)
           fvaltmp = fvaltmp + costtmp/100000000.0
           fval = fval + fvaltmp
           bias = 0.0

*          If Pr limit is violated.

           if (costtmp .ge. 1.0e100) then
*               iv = iv + 1
              print *, 'Pr limit violated...', Pr1
           else
*               ic = ic + 1
*              print *, 'Cool !... '
           end if
 1122  continue
*
*           Evaluating fitness values
*
            fitness(ichr) = 1000000.0 / fval
            fitave = fitave + fitness(ichr)

            if (fitness(ichr) .gt. fitmax) then
               fitmax = fitness(ichr)
            end if

            if (fitness(ichr) .gt. bestfit) then
*              print *, fval
               station = 0
               bestfit = fitness(ichr)
               bestfval = fval
               do 55 m = 1, szind
                  bestind(m) = pop(ichr, m)
   55          continue
               do 60 n = 1, maxgen*intval
                  bestx(n) = poppow(ichr, n)
   60          continue
            end if
   25    continue

*        End ichr

         fitave = fitave / real(szpop)
         write (4, 666) fitave, bestfit
  666    format (f20.5, f20.5)

*         print *, 'Fitness scaling ...'
*
*        -------------  Sigma Truncation Scaling --------------
*        f = f - (fave - c*sigma)
*        The constant c is chosen as a reasonable muliple of the
*        population standard deviation (between 1 and 3) and
*        negative result aaare arbitrarily set to 0.
*
         c = 2.0
         do 61 i = 1, szpop
            fitdev = fitdev + (fitness(i) - fitave)**2
   61    continue
         fitdev = fitdev**0.5
```

```
         do 62 i = 1, szpop
            fitnessn(i) = fitness(i) - fitave + c*fitdev
            if (fitnessn(i) .lt. 0.0) fitness(i) = 0.0
   62    continue

*
*        Performing Tournament selection.
*
         do 65 ichr = 1, szpop, szchild
            do 70 ichild = 1, szchild
               if ((ipick+sztour) .gt. szpop) then
*                  Shuffles population randomly.
                  ipick = 1
                  do 75 m = 1, szpop-1
                     call ran3(1, val)
                     i = m + dint(dble(szpop-m)*val) + 1
                     do 80 n = 1, maxgen
                        j = pop(i, n)
                        pop(i, n) = pop(m, n)
                        pop(m, n) = j
   80                continue
                     rtmp = fitness(i)
                     fitness(i) = fitness(m)
                     fitness(m) = rtmp
                     rtmp = fitnessn(i)
                     fitnessn(i) = fitnessn(m)
                     fitnessn(m) = rtmp
   75             continue
               end if

               mate(ichild) = ipick
               do 85 i = 1, sztour-1
                  if (fitnessn(mate(ichild)) .lt. fitnessn(ipick+i))
     +             then
                     mate(ichild) = ipick + i
                  end if
   85          continue
               ipick = ipick + sztour
   70       continue
*
*           Creates child chromosomes.
*
            do 90 n = 1, szind
               popc(ichr, n) = pop(mate(1), n)
               popc(ichr+1, n) = pop(mate(2), n)
   90       continue
*
*           Performing uniform crossover.
*
*            print *, 'Crossing over ...'
            if (fitness(mate(1)) .gt. fitness(mate(2))) then
               if (fitness(mate(1)) .ge. fitave) then
                  pcross = k1*(fitmax-fitness(mate(1)))/(fitmax-fitave)
               else
                  pcross = k3
               end if
            else
               if (fitness(mate(2)) .ge. fitave) then
                  pcross = k1*(fitmax-fitness(mate(2)))/(fitmax-fitave)
               else
                  pcross = k3
               end if
            end if
            do 95 n = 1, szind
               call ran3(1, val)
               if (val .lt. pcross) then
                  popc(ichr, n) = pop(mate(2), n)
                  popc(ichr+1, n) = pop(mate(1), n)
               end if
   95       continue

*        End ichr (selection)
*
*        Performing random mutation.
*
*        print *, 'Mutating ...'
*
*        Performing Jump mutation.
*
```

```
         pmutate = real(station)/real(stamax)*pmumax
         call ran3(1, val)
         if (val .lt. pmutate) then
            do 6006 iperiod = 1, intval
             ii = 0
              do 6007 i = 1, maxgen-1
               if ((i .ge. 1) .and. (i .le. 4)) then
                   goto 6007
                  end if

               if ((i .ge. 11) .and. (i .le. 18)) then
                   goto 6007
               end if

               if (i .eq. 20) then
                   goto 6007
               end if

               if ((i .ge. 27) .and. (i .le. 57)) then
                   goto 6007
               end if

               if ((i .ge. 60) .and. (i .le. 70)) then
                   goto 6007
               end if

                 ii = ii + 1
                 call ran3(1, val)
                 if (val .lt. 0.5) then
                    popcx(ichr, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr, ((iperiod-1)*(maxgen) + i)) -
     + 2**-szchr*(mwmax(i) - mwmin(i))
                 else
                   popcx(ichr, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr, ((iperiod-1)*(maxgen) + i)) +
     + 2**-szchr*(mwmax(i) - mwmin(i))
                end if
          tmp = (popcx(ichr, ((iperiod-1)*(maxgen) + i))- mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 6008 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 1
                      tmp = tmp - 2.0**-m
                   else
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 6008             continue
*                 Re-engraying...
                 do 6009 m = 2, szchr
          itmp = popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +   + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 6009             continue
                 do 6010 m = 2, szchr
          popc(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +      = chrtmp(m)
 6010             continue
 6007         continue
 6006       continue
          end if

          call ran3(1, val)
          if (val .lt. pmutate) then
             do 6012 iperiod = 1, intval
               ii = 0
               do 6013 i = 1, maxgen-1
                if ((i .ge. 1) .and. (i .le. 4)) then
                    goto 6013
                   end if

                if ((i .ge. 11) .and. (i .le. 18)) then
                    goto 6013
                end if

                if (i .eq. 20) then
                    goto 6013
```

```
              end if

              if ((i .ge. 27) .or. (i .le. 57)) then
                  goto 6013
              end if

              if ((i .ge. 60) .or. (i .le. 70)) then
                  goto 6013
              end if

                ii = ii + 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr+1, ((iperiod-1)*(damaxgen) + ii)) -
     + 2**-szchr*(mwmax(i) - mwmin(i))
                 else
                   popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
     + = popcx(ichr+1, ((iperiod-1)*(damaxgen) + ii)) +
     + 2**-szchr*(mwmax(i) - mwmin(i))
                end if
         tmp = (popcx(ichr+1, ((iperiod-1)*(maxgen) + i))- mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 6014 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 1
                      tmp = tmp - 2.0**-m
                   else
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 6014            continue
*                Re-engraying...
                do 6015 m = 2, szchr
       itmp = popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr+1,((iperiod-1)*(damaxgen-1)
     + *szchr + (szchr*(ii-1)+m-1)))
                itmp = itmp-(itmp/2)*2
                chrtmp(m) = itmp
 6015            continue
                do 6016 m = 2, szchr
       popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) = chrtmp(m)
 6016            continue
 6013         continue
 6012      continue
        end if
*-----------------------------------------------------
*       Deterministic Crowding
*-----------------------------------------------------
*           print *, '------- IChr I ---------'
            fval = 0.0
         do 8201 iperiod = 1, intval
*        print *, '###### Interval : ', iperiod, ' #######'
             bias = 0.0
            trapped = 0
            if (iperiod .eq. 1) then
             do 8202 i = 1, maxgen
                tmp = mwnow(i) - rampdn(i)
                call chkmw(tmp, lowlim(i), hilim(i))
                mwmin(i) = tmp
                tmp = mwnow(i) + rampup(i)
                call chkmw(tmp, lowlim(i), hilim(i))
                mwmax(i) = tmp
*               print *,
*     + mwmin(i), mwmax(i)
 8202        continue
            else
             do 8203 i = 1, maxgen
                tmp = popcx(ichr, ((iperiod-2)*(maxgen) + i))
     +          - rampdn(i)
                call chkmw(tmp, lowlim(i), hilim(i))
                mwmin(i) = tmp
                tmp = popcx(ichr, ((iperiod-2)*(maxgen) + i))
     +          + rampup(i)
                call chkmw(tmp, lowlim(i), hilim(i))
                mwmax(i) = tmp
*               print *,
*     + mwmin(i), popcx(ichr, ((iperiod-2)*(maxgen) + i)), mwmax(i)
```

```
 8203       continue
           end if

              opmax = 0.0
              opmin = 0.0
              mustrunmw = 0.0
              do 8331 j = 1, maxgen-1
                 if ((j .ge. 1) .and. (j .le. 4)) then
                    mustrunmw = mustrunmw + mwmax(j)
                    goto 8331
                 end if
                  if ((j .ge. 11) .and. (j .le. 18)) then
                    mustrunmw = mustrunmw + mwmax(j)
                    goto 8331
                 end if
                  if (j .eq. 20) then
                    mustrunmw = mustrunmw + mwmax(j)
                     goto 8331
                 end if
                  if ((j .ge. 27) .and. (j .le. 57)) then
                    mustrunmw = mustrunmw + mwmax(j)
                     goto 8331
                 end if
                  if ((j .ge. 60) .and. (j .le. 70)) then
                    mustrunmw = mustrunmw + mwmax(j)
                     goto 8331
                 end if

                   opmax = opmax + mwmax(j)
                   opmin = opmin + mwmin(j)
 8331          continue

 8224       sumpowtmp = 0.0
            fvaltmp = 0.0
*           Check edemand
            pmax = opmax
            pmin = opmin
            remain = loadf(iperiod)*odemand+osysloss*loadf(iperiod)
     + -  (mwmax(maxgen)+mwmin(maxgen))/2.0 - mustrunmw
            forceup = .false.
            forcedn = .false.

            ii = 0
            do 8204 i = 1, maxgen-1

*               Check dispatchable gernerating units
                if ((i .ge. 1) .and. (i .le. 4)) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8217
                end if

                if ((i .ge. 11) .and. (i .le. 18)) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8217
                end if

                if (i .eq. 20) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8217
                end if

                if ((i .ge. 27) .and. (i .le. 57)) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8217
                end if

                if ((i .ge. 60) .and. (i .le. 70)) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8217
                end if

                ii = ii + 1
                 if (forceup) then
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                  do 8205 m = 1, szchr
*                     Engraying...
                     if (m .eq. 1) then
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +    = 1
                     else
```

```
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +    = 0
                   end if
 8205             continue

           remain = remain - popcx(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)

                goto 8217
              end if

              if (forcedn) then
                popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
                do 8206 m = 1, szchr
*                   Engraying...
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +     = 0
 8206             continue
           remain = remain - popcx(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)

                goto 8217
              end if

          if (bias .ne. 0.0) then
           tmp = popcx(ichr, ((iperiod-1)*(maxgen) + i)) + bias

       if (tmp .gt. mwmax(i)) then
         bias = bias - mwmax(i) +
     +                 popcx(ichr, ((iperiod-1)*(maxgen) + i))
          popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
       else if (tmp .lt. mwmin(i)) then
         bias = bias - mwmin(i) +
     +                 popcx(ichr, ((iperiod-1)*(maxgen) + i))
          popcx(ichr, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
       else
          popcx(ichr, ((iperiod-1)*(maxgen) + i)) = tmp
           bias = 0.0
       end if
*                Re-encoding...
       tmp = (popcx(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 8207 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +   (szchr*(ii-1)+szchr+1-m))) = 1
                      tmp = tmp - 2.0**-m
                   else
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +   (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 8207             continue
*                 Re-engraying...
                do 8208 m = 2, szchr
       itmp = popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) +popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m-1)))
                itmp = itmp-(itmp/2)*2
                chrtmp(m) = itmp
 8208            continue
                do 8209 m = 2, szchr
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 8209            continue

         else
              popcx(ichr, ((iperiod-1)*(maxgen) + i)) = 0.0
              itmp = 0
*              Degraying
              do 8210 j = 1, szchr
           itmp = itmp + popc(ichr,
     + ((iperiod-1)*(damaxgen-1)*szchr+(szchr*(ii-1)+j)))
                itmp = itmp-(itmp/2)*2
                popcx(ichr, ((iperiod-1)*(maxgen) + i)) =
     + popcx(ichr, ((iperiod-1)*(maxgen) + i)) +
     + real(itmp)*2.0**-(szchr-j+1)
 8210          continue
                popcx(ichr, ((iperiod-1)*(maxgen) + i)) =
```

```
     + mwmin(i) + popcx(ichr, ((iperiod-1)*(maxgen) + i))*
     + (mwmax(i) - mwmin(i))
           end if
           remain = remain - popcx(ichr, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)
*
*    ------------------------------------------------------
*
           if (ii .lt. damaxgen-1) then
              if (remain .ge. pmax) then
                 forceup = .true.
       remain = remain + popcx(ichr, ((iperiod-1)*(maxgen) + i))
       popcx(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmax
*       if ((popcx(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (popcx(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     +  then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
*       print *, 'remain .ge. pmax ->', remain
       remain = remain - popcx(ichr, ((iperiod-1)*(maxgen) + i))
*                Re-encoding...
*       print *, 'remain (new) ->', remain
       tmp = (popcx(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 8211 m = 1, szchr
                   if (tmp .gt. 2.0**-m) then
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                (szchr*(ii-1)+szchr+1-m))) = 1
                     tmp = tmp - 2.0**-m
                   else
        popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                (szchr*(ii-1)+szchr+1-m))) = 0
                   end if
 8211              continue
*                  Re-engraying...
                  do 8212 m = 2, szchr
       itmp = popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 8212              continue
                  do 8213 m = 2, szchr
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 8213              continue
                 end if
               if (remain .le. pmin) then
                  forcedn = .true.
       remain = remain + popcx(ichr, ((iperiod-1)*(maxgen) + i))
*       print *, 'In first:', popcx(ichr, ((iperiod-1)*(maxgen) + i))
*     + , mwmax(i), mwmin(i), remain
                  popcx(ichr, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmin
*       if ((popcx(ichr,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (popcx(ichr,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     + then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
          remain = remain - popcx(ichr, ((iperiod-1)*(maxgen) + i))
*                Re-encoding...
       tmp = (popcx(ichr, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
*          print *, 'check ', popcx(ichr, ((iperiod-1)*(maxgen) + i)),
*     +             mwmin(i), remain
                  do 8214 m = 1, szchr
                     if (tmp .gt. 2.0**-m) then
         popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 1
                       tmp = tmp - 2.0**-m
                     else
         popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +                     (szchr*(ii-1)+szchr+1-m))) = 0
                     end if
 8214              continue
*                  Re-engraying...
```

```
                do 8215 m = 2, szchr
        itmp = popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) +
     +   popc(ichr,((iperiod-1)*(damaxgen-1)*szchr +
     +   (szchr*(ii-1)+m-1)))
                itmp = itmp-(itmp/2)*2
                chrtmp(m) = itmp
 8215            continue
                do 8216 m = 2, szchr
       popc(ichr,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     + = chrtmp(m)
 8216            continue
              end if
           end if

*
*              Check power participation
*

 8217       sumpowtmp = sumpowtmp + popcx(ichr,
     + ((iperiod-1)*(maxgen) + i))

*               print *, 'Ind index = ', ichr, ' Gen = ', iter,
*     + 'Period = ', iperiod
*               print *, 'Pgen', i, ' = ', popcx(ichr,
*     + ((iperiod-1)*(maxgen) + i)), ' remain = ', remain, ' pmax = ',
*     + pmax, ' pmin = ', pmin
*               print *, 'forceup = ', forceup, ' forcedn = ', forcedn
*               print *, 'sumpowtmp = ', sumpowtmp
*               pause

              costtmp = heatrate(hrfn(i),
     + popcx(ichr, ((iperiod-1)*(maxgen) + i)), lowlim(i),
     + hilim(i))
*              fval (MBaht/h)
              fvaltmp = fvaltmp + costtmp/100000000.0
*             print *, 'Total cost =', fvaltmp
 8204       continue
*
*     Determine Pr
*          Demand - SumMW + Ploss - Pr = 0
*          Ploss = aPr^2 + bPr + c
*              a = Brr
*              b = BraPa + PatBar + B0r
*              c = PatBaaPa + PatB0a + B00
*          Thus, modify coefficients.
*              a = Brr
*              b = b-1 = BraPa + PatBar + B0r - 1
*              c = PatBaaPa + PatB0a + B00 + Demand - sumMW
*          Determines Pr
*             Pr = -b +- sqrt(b^2 - 4*a*c)
*                  ----------------------
*                           2*a
           coef_b = 0.0
           coef_c = 0.0

           do 8218 ii = 1, maxgen-1
              coef_b = coef_b +
     + bcoeff(maxgen, ii)*popcx(ichr, ((iperiod-1)*(maxgen-1) +
     + ii))/100.0
              tempb(ii) = 0.0
 8218      continue
           coef_b = coef_b*2.0 + b0(maxgen) - 1.0

           do 8219 ii = 1, maxgen-1
              do 8220 iii = 1, maxgen-1
                 tempb(ii) = tempb(ii) +
     + popcx(ichr, ((iperiod-1)*(maxgen)+iii))*bcoeff(iii, ii)/100.0
 8220            continue
 8219      continue
           do 8221 ii = 1, maxgen-1
              coef_c = coef_c + tempb(ii)*
     + popcx(ichr, ((iperiod-1)*(maxgen) + ii))/100.0
 8221      continue
           do 8222 ii = 1, maxgen-1
              coef_c = coef_c + popcx(ichr,((iperiod-1)*(maxgen)+ii))
     +        *b0(ii)/100.0
 8222      continue
       coef_c = coef_c + b00 + (loadf(iperiod)*odemand-sumpowtmp)/
     +    100.0
```

```
*           print *, 'load', loadf(iperiod)*odemand, sumpowtmp
            Pr1 = -1.0*coef_b + sqrt(coef_b*coef_b -
     +                        4.0*bcoeff(maxgen,maxgen)*coef_c)
            Pr1 = 100.0*Pr1/(2.0*bcoeff(maxgen,maxgen))
            Pr2 = -1.0*coef_b - sqrt(coef_b*coef_b -
     +                        4.0*bcoeff(maxgen,maxgen)*coef_c)
            Pr2 = 100.0*Pr2/(2.0*bcoeff(maxgen,maxgen))

            if (Pr1 .le. 0.0) then
               if (Pr2 .gt. 0.0) Pr1 = Pr2
            end if
            if (Pr1 .gt. Pr2) Pr1 = Pr2
            sumpowtmp = sumpowtmp + Pr1
            popcx(ichr, (iperiod-1)*maxgen + maxgen) = Pr1

            if (Pr1 .gt. mwmax(maxgen)) then
*               print *, 'Power imbalance occurred. (Pr1 > mwmax)',
*     +              Pr1, '>', mwmax(maxgen),
*     +        '(IChr = ', ichr, ' Period = ', iperiod, ')'
              if (trapped .gt. trapmax) then
                 print *, 'Trapped'
*               Re-building the chromosome
                do 8223 ii = 1, szchr*(damaxgen-1)
                   popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +             + ii)) = 1
                   call ran3(1, val)
                   if (val .lt. 0.5) then
                      popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +             + ii)) = 0
                   end if
 8223           continue
               trapped = 0
               goto 8224
             end if
               bias = Pr1 - mwmax(maxgen)
*              Loss compensating...
*              bias = bias + bias*sysloss/sumpowtmp*biasac
*              print *, 'bias = ', bias, sysloss
               trapped = trapped + 1
               goto 8224
            end if
            if (Pr1 .lt. mwmin(maxgen)) then
*               print *, 'Power imbalance occurred. (Pr1 < mwmin)',
*     +        Pr1, '<', mwmin(maxgen),
*     +        '(IChr = ', ichr, ' Period = ', iperiod, ')'
              if (trapped .gt. trapmax) then
                 print *, 'Trapped'
*               Re-building the chromosome
                print *, 'Trapped'
                do 8225 ii = 1, szchr*(damaxgen-1)
                   popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +             + ii)) = 1
                   call ran3(1, val)
                   if (val .lt. 0.5) then
                      popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +             + ii)) = 0
                   end if
8225           continue
               trapped = 0
               goto 8224
             end if
               bias = Pr1 - mwmin(maxgen)
*              Loss compensating...
               bias = bias + bias*sysloss/sumpowtmp*biasac
*              print *, 'bias = ', bias, sysloss
               trapped = trapped + 1
*              print *, trapped
               goto 8224
            end if
            costtmp = heatrate(hrfn(maxgen), Pr1, mwmin(maxgen),
     + mwmax(maxgen))
            sumpow = sumpowtmp
*           fval (MBaht)
            fvaltmp = fvaltmp + costtmp/100000000.0
            fval = fval + fvaltmp
            bias = 0.0

*           If Pr limit is violated.

            if (costtmp .ge. 1.0e100) then
```

```
*               iv = iv + 1
              print *, 'Pr limit violated...', Pr1
            else
*               ic = ic + 1
*              print *, 'Cool !... '
            end if
 8201   continue
*
*            Evaluating fitness values
*
            fitc1 = 1000000.0 / fval
*=============================================================
*            print *, '------- IChr II ---------'

            fval = 0.0
         do 8301 iperiod = 1, intval
*        print *, '###### Interval : ', iperiod, ' #######'
            bias = 0.0
            trapped = 0
            if (iperiod .eq. 1) then
            do 8302 i = 1, maxgen
               tmp = mwnow(i) - rampdn(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmin(i) = tmp
               tmp = mwnow(i) + rampup(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmax(i) = tmp
*              print *,
*     + mwmin(i), mwmax(i)

 8302       continue
            else
            do 8303 i = 1, maxgen
               tmp = popcx(ichr+1, ((iperiod-2)*(maxgen) + i))
     +         - rampdn(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmin(i) = tmp
               tmp = popcx(ichr+1, ((iperiod-2)*(maxgen) + i))
     +         + rampup(i)
               call chkmw(tmp, lowlim(i), hilim(i))
               mwmax(i) = tmp
*              print *,
*     + mwmin(i), popcx(ichr+1, ((iperiod-2)*(maxgen) + i)), mwmax(i)
 8303       continue
            end if

               opmax = 0.0
               opmin = 0.0
               mustrunmw = 0.0
               do 9331 j = 1, maxgen-1
                  if ((j .ge. 1) .and. (j .le. 4)) then
                     mustrunmw = mustrunmw + mwmax(j)
                     goto 9331
                  end if
                  if ((j .ge. 11) .and. (j .le. 18)) then
                     mustrunmw = mustrunmw + mwmax(j)
                     goto 9331
                  end if
                  if (j .eq. 20) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 9331
                  end if
                  if ((j .ge. 27) .and. (j .le. 57)) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 9331
                  end if
                  if ((j .ge. 60) .and. (j .le. 70)) then
                     mustrunmw = mustrunmw + mwmax(j)
                      goto 9331
                  end if

                    opmax = opmax + mwmax(j)
                    opmin = opmin + mwmin(j)
 9331           continue

 8324       sumpowtmp = 0.0
            fvaltmp = 0.0
*           Check edemand
            pmax = opmax
            pmin = opmin
```

```
            remain = loadf(iperiod)*odemand+osysloss*loadf(iperiod)
     + -  (mwmax(maxgen)+mwmin(maxgen))/2.0 - mustrunmw
            forceup = .false.
            forcedn = .false.

            ii = 0
            do 8304 i = 1, maxgen-1
*               Check dispatchable gernerating units
                if ((i .ge. 1) .and. (i .le. 4)) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8317
                end if

                if ((i .ge. 11) .and. (i .le. 18)) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                 goto 8317
                end if

                if (i .eq. 20) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8317
                end if

                if ((i .ge. 27) .and. (i .le. 57)) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8317
                end if

                if ((i .ge. 60) .and. (i .le. 70)) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                    goto 8317
                end if

                ii = ii + 1

                 if (forceup) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
                  do 8305 m = 1, szchr

*                     Engraying...
                     if (m .eq. 1) then
       popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +    = 1
                      else
       popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +    = 0
                    end if
 8305             continue
*          print *, '(1)',
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+1))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+2))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+3))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+4))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+5))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+6))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+7))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+8))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+9))),
*     + popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+10))),
*     + ' -> ', 'u', i, mwmin(i), mwmax(i),
*     + popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
                remain = remain - popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
                pmax = pmax - mwmax(i)
                pmin = pmin - mwmin(i)

                  goto 8317
                end if

                if (forcedn) then
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
                  do 8306 m = 1, szchr
*                     Engraying...
      popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +  = 0
 8306             continue
                remain = remain - popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
                pmax = pmax - mwmax(i)
                pmin = pmin - mwmin(i)

                  goto 8317
```

```
              end if

          if (bias .ne. 0.0) then
        tmp = popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) + bias

       if (tmp .gt. mwmax(i))
     +   then
         bias = bias - mwmax(i) +
     +                popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
          popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmax(i)
       else if (tmp .lt. mwmin(i)) then
         bias = bias - mwmin(i) +
     +                popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
          popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = mwmin(i)
       else
           popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = tmp
           bias = 0.0
       end if
*                Re-encoding...
       tmp = (popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                do 8307 m = 1, szchr
                  if (tmp .gt. 2.0**-m) then
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +  (szchr*(ii-1)+szchr+1-m))) = 1
                    tmp = tmp - 2.0**-m
                  else
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +   (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 8307             continue
*                 Re-engraying...
                 do 8308 m = 2, szchr
       itmp = popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr+1,((iperiod-1)*(damaxgen-1)
     +  *szchr + (szchr*(ii-1)+m-1)))
                 itmp = itmp-(itmp/2)*2
                 chrtmp(m) = itmp
 8308             continue
                 do 8309 m = 2, szchr
       popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr + (szchr*(ii-1)+m)))
     +     = chrtmp(m)
 8309            continue
          else
               popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) = 0.0
               itmp = 0
*              Degraying
               do 8310 j = 1, szchr
           itmp = itmp + popc(ichr+1,
     + ((iperiod-1)*(damaxgen-1)*szchr+(szchr*(ii-1)+j)))
                itmp = itmp-(itmp/2)*2
               popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) =
     + popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) +
     + real(itmp)*2.0**-(szchr-j+1)
 8310          continue
                popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) =
     + mwmin(i) + popcx(ichr+1, ((iperiod-1)*(maxgen) + i))*
     + (mwmax(i) - mwmin(i))
          end if
          remain = remain - popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
              pmax = pmax - mwmax(i)
              pmin = pmin - mwmin(i)
*
*    ------------------------------------------------------
*
          if (ii .lt. damaxgen-1) then
              if (remain .ge. pmax) then
                 forceup = .true.
       remain = remain + popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
       popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmax
*       if ((popcx(ichr+1,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (popcx(ichr+1,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     +  then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
*       print *, 'remain .ge. pmax ->', remain
       remain = remain - popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
*                Re-encoding...
```

```
*       print *, 'remain (new) ->', remain
        tmp = (popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
                  do 8311 m = 1, szchr
                     if (tmp .gt. 2.0**-m) then
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                  (szchr*(ii-1)+szchr+1-m))) = 1
                     tmp = tmp - 2.0**-m
                  else
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                  (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 8311             continue
*                 Re-engraying...
                  do 8312 m = 2, szchr
        itmp = popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) + popc(ichr+1,((iperiod-1)*(damaxgen-1)
     + *szchr + (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 8312             continue
                  do 8313 m = 2, szchr
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     +     = chrtmp(m)
 8313             continue
                 end if
               if (remain .le. pmin) then
                  forcedn = .true.
        remain = remain + popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
*       print *, 'In first:', popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
*     + , mwmax(i), mwmin(i), remain
                  popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) =
     + remain - pmin
*       if ((popcx(ichr+1,((iperiod-1)*(maxgen) + i)) .gt. mwmax(i))
*     + .or. (popcx(ichr+1,((iperiod-1)*(maxgen) + i)) .lt. mwmin(i)))
*     + then
*          print *, 'Please check the possible power generation range',
*     + ': unit', i
*       end if
          remain = remain - popcx(ichr+1, ((iperiod-1)*(maxgen) + i))
*                 Re-encoding...
        tmp = (popcx(ichr+1, ((iperiod-1)*(maxgen) + i)) - mwmin(i))/
     + (mwmax(i) - mwmin(i))
*          print *, 'check ', popcx(ichr+1,((iperiod-1)*(maxgen) + i)),
*     +             mwmin(i), remain
                  do 8314 m = 1, szchr
                     if (tmp .gt. 2.0**-m) then
         popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                       (szchr*(ii-1)+szchr+1-m))) = 1
                     tmp = tmp - 2.0**-m
                  else
         popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +                       (szchr*(ii-1)+szchr+1-m))) = 0
                  end if
 8314             continue
*                 Re-engraying...
                  do 8315 m = 2, szchr
         itmp = popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     + (szchr*(ii-1)+m))) +
     +   popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +
     +  (szchr*(ii-1)+m-1)))
                  itmp = itmp-(itmp/2)*2
                  chrtmp(m) = itmp
 8315             continue
                  do 8316 m = 2, szchr
        popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr +(szchr*(ii-1)+m)))
     + = chrtmp(m)
 8316             continue
               end if
            end if

*
*              Check power participation
*

 8317       sumpowtmp = sumpowtmp + popcx(ichr+1,
     + ((iperiod-1)*(maxgen) + i))

*               print *, 'Ind index = ', ichr+1, ' Gen = ', iter,
*     + 'Period = ', iperiod
```

```
*               print *, 'Pgen', i, ' = ', popcx(ichr+1,
*     + ((iperiod-1)*(maxgen) + i)), ' remain = ', remain, ' pmax = ',
*     + pmax, ' pmin = ', pmin
*               print *, 'forceup = ', forceup, ' forcedn = ', forcedn
*               print *, 'sumpowtmp = ', sumpowtmp
*               pause

               costtmp = heatrate(hrfn(i),
     + popcx(ichr+1, ((iperiod-1)*(maxgen) + i)), lowlim(i),
     + hilim(i))
*              fval (MBaht/h)
               fvaltmp = fvaltmp + costtmp/100000000.0
*              print *, 'Total cost =', fvaltmp
 8304       continue
*
*     Determine Pr
*          Demand - SumMW + Ploss - Pr = 0
*          Ploss = aPr^2 + bPr + c
*              a = Brr
*              b = BraPa + PatBar + B0r
*              c = PatBaaPa + PatB0a + B00
*          Thus, modify coefficients.
*              a = Brr
*              b = b-1 = BraPa + PatBar + B0r - 1
*              c = PatBaaPa + PatB0a + B00 + Demand - sumMW
*          Determines Pr
*             Pr = -b +- sqrt(b^2 - 4*a*c)
*                  ----------------------
*                             2*a
           coef_b = 0.0
           coef_c = 0.0

           do 8318 ii = 1, maxgen-1
              coef_b = coef_b +
     + bcoeff(maxgen, ii)*popcx(ichr+1, ((iperiod-1)*(maxgen-1) +
     + ii))/100.0
              tempb(ii) = 0.0
 8318      continue
           coef_b = coef_b*2.0 + b0(maxgen) - 1.0

           do 8319 ii = 1, maxgen-1
              do 8320 iii = 1, maxgen-1
                 tempb(ii) = tempb(ii) +
     + popcx(ichr+1, ((iperiod-1)*(maxgen)+iii))*bcoeff(iii, ii)/100.0
 8320         continue
 8319      continue
           do 8321 ii = 1, maxgen-1
              coef_c = coef_c + tempb(ii)*
     + popcx(ichr+1, ((iperiod-1)*(maxgen) + ii))/100.0
 8321      continue
           do 8322 ii = 1, maxgen-1
              coef_c = coef_c + popcx(ichr+1,((iperiod-1)*(maxgen)+ii))
     +        *b0(ii)/100.0
 8322      continue
       coef_c = coef_c + b00 + (loadf(iperiod)*odemand-sumpowtmp)/

     +    100.0
*            print *, 'load', loadf(iperiod)*odemand, sumpowtmp
           Pr1 = -1.0*coef_b + sqrt(coef_b*coef_b -
     +                          4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr1 = 100.0*Pr1/(2.0*bcoeff(maxgen,maxgen))
           Pr2 = -1.0*coef_b - sqrt(coef_b*coef_b -
     +                          4.0*bcoeff(maxgen,maxgen)*coef_c)
           Pr2 = 100.0*Pr2/(2.0*bcoeff(maxgen,maxgen))

           if (Pr1 .le. 0.0) then
              if (Pr2 .gt. 0.0) Pr1 = Pr2
           end if
           if (Pr1 .gt. Pr2) Pr1 = Pr2
           sumpowtmp = sumpowtmp + Pr1
           popcx(ichr+1, (iperiod-1)*maxgen + maxgen) = Pr1

           if (Pr1 .gt. mwmax(maxgen)) then
*               print *, 'Power imbalance occurred. (Pr1 > mwmax)',
*     +             Pr1, '>', mwmax(maxgen),
*     +       '(IChr = ', ichr+1, ' Period = ', iperiod, ')'
             if (trapped .gt. trapmax) then
                 print *, 'Trapped'
*              Re-building the chromosome
               do 8323 ii = 1, szchr*(damaxgen-1)
```

```
                popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   popc(ichr+1,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 0
                 end if
 8323          continue
              trapped = 0
              goto 8324
             end if
               bias = Pr1 - mwmax(maxgen)
*              Loss compensating...
               bias = bias + bias*sysloss/sumpowtmp*biasac
*              print *, 'bias = ', bias, sysloss
               trapped = trapped + 1
               goto 8324
            end if
            if (Pr1 .lt. mwmin(maxgen)) then
*               print *, 'Power imbalance occurred. (Pr1 < mwmin)',
*     +       Pr1, '<', mwmin(maxgen),
*     +       '(IChr = ', ichr, ' Period = ', iperiod, ')'

             if (trapped .gt. trapmax) then
                  print *, 'Trapped'
*              Re-building the chromosome
               do 8325 ii = 1, szchr*(damaxgen-1)
                  popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 1
                call ran3(1, val)
                if (val .lt. 0.5) then
                   popc(ichr,((iperiod-1)*(damaxgen-1)*szchr
     +          + ii)) = 0
                 end if
8325          continue
              trapped = 0
              goto 8324
             end if
               bias = Pr1 - mwmin(maxgen)
*              Loss compensating...
               bias = bias + bias*sysloss/sumpowtmp*biasac
*              print *, 'bias = ', bias, sysloss
               trapped = trapped + 1
*              print *, trapped
               goto 8324
            end if
            costtmp = heatrate(hrfn(maxgen), Pr1, mwmin(maxgen),
     + mwmax(maxgen))
            sumpow = sumpowtmp
*           fval (MBaht)
            fvaltmp = fvaltmp + costtmp/100000000.0
            fval = fval + fvaltmp
            bias = 0.0

*           If Pr limit is violated.


            if (costtmp .ge. 1.0e100) then
*                iv = iv + 1
               print *, 'Pr limit violated...', Pr1
            else
*               ic = ic + 1
*              print *, 'Cool !... '
            end if
 8301  continue
*
*            Evaluating fitness values
*
             fitc2 = 1000000.0 / fval

             dp1c1 = ((poppow(ichr, 1) - popcx(ichr, 1))**2 +
     +               (poppow(ichr, 2) - popcx(ichr, 2))**2)**0.5
             dp2c2 = ((poppow(ichr+1, 1) - popcx(ichr+1, 1))**2 +
     +               (poppow(ichr+1, 2) - popcx(ichr+1, 2))**2)**0.5
             dp1c2 = ((poppow(ichr, 1) - popcx(ichr+1, 1))**2 +
     +               (poppow(ichr, 2) - popcx(ichr+1, 2))**2)**0.5
             dp2c1 = ((poppow(ichr+1, 1) - popcx(ichr, 1))**2 +
     +               (poppow(ichr+1, 2) - popcx(ichr, 2))**2)**0.5

             if ((dp1c1+dp2c2) .le. (dp1c2+dp2c1)) then
```

```
               if (fitc1 .gt. fitness(mate(1))) then
                  do 5701 m = i, szind
                     pop(mate(1), m) = popc(ichr, m)
 5701             continue
                end if
                if (fitc2 .gt. fitness(mate(2))) then
                   do 5702 m = i, szind
                      pop(mate(2), m) = popc(ichr+1, m)
 5702             continue
                end if
            else
                if (fitc2 .gt. fitness(mate(1))) then
                   do 5703 m = i, szind
                      pop(mate(1), m) = popc(ichr+1, m)
 5703             continue
                end if
                if (fitc1 .gt. fitness(mate(2))) then
                   do 5704 m = i, szind
                      pop(mate(2), m) = popc(ichr, m)
 5704             continue
                end if
           end if

  65    continue
*
*        Replace new generation with elitism.
*
*         print *, 'Regenerating...'

         do 110 m = 1, szpop
            do 115 n = 1, szind
               pop(m, n) = popc(m, n)
  115       continue
  110    continue
*
*        Checks population homogeneity
*
*         do 125 i = 1, szind
*            numzero(i) = 0
*  125    continue
*         tmp = 0.0
*         do 130 i = 1, szind
*            do 135 j = 1, szpop
*               if (pop(j, i) .eq. 0) then
*                  numzero(i) = numzero(i) + 1
*               end if
*  135       continue
*            if (numzero(i) .ge. (szpop/2)) then
*               tmp = tmp + real(numzero(i)) / real(szpop)
*            else
*               tmp = tmp + real(szpop - numzero(i)) / real(szpop)
*            end if
*  130    continue
*         tmp = tmp / real(szind)
*          print *, 'best heat = ', bestheat, ' and power diff = ',
*       + (bestgen - loadf(iperiod)*odemand - bestloss)
*          print *, '% Homogenity = ', tmp
*
*        Checks termination criterion
*
         station = station + 1
*         if ((tmp .gt. hmlevel) .or. (station .gt. 250)) goto 500
          if (station .gt. stamax) goto 500

  920  continue

*       End generation
*     print *, 'Generation complete !'
500     total = etime(elapsed)
        write (3, *) bestx(1), bestx(2), bestx(3), bestx(4),
     + bestx(5), bestx(6), bestx(7), bestx(8), bestx(9), bestx(10),
     + bestx(11), bestx(12), bestx(13), bestx(14), bestx(15),
     + bestx(16), bestx(17), bestx(18), bestx(19), bestx(20),
     + bestx(21), bestx(22), bestx(23), bestx(24), bestx(25),
     + bestx(26), bestx(27), bestx(28), bestx(29), bestx(30),
     + bestx(31), bestx(32), bestx(33), bestx(34), bestx(35),
     + bestx(36), bestx(37), bestx(38), bestx(39), bestx(40),
     + bestx(41), bestx(42), bestx(43), bestx(44), bestx(45),
     + bestx(46), bestx(47), bestx(48), bestx(49), bestx(50),
     + bestx(51), bestx(52), bestx(53), bestx(54), bestx(55),
```

```
+ bestx(56), bestx(57), bestx(58), bestx(59), bestx(60),
+ bestx(61), bestx(62), bestx(63), bestx(64), bestx(65),
+ bestx(66), bestx(67), bestx(68), bestx(69), bestx(70),
+ bestx(71), bestx(72), bestx(73), bestx(74), bestx(75),
+ bestx(76), bestx(77), bestx(78), bestx(79), bestx(80),
+ bestx(81), bestx(82), bestx(83), bestx(84), bestx(85),
+ bestx(86), bestx(87), bestx(88), bestx(89), bestx(90),
+ bestx(91), bestx(92), bestx(93), bestx(94), bestx(95),
+ bestx(96), bestx(97), bestx(98), bestx(99), bestx(100),
+ bestx(101), bestx(102), bestx(103), bestx(104), bestx(105),
+ bestx(106), bestx(107), bestx(108), bestx(109), bestx(110),
+ bestx(111), bestx(112), bestx(113), bestx(114), bestx(115),
+ bestx(116), bestx(117), bestx(118), bestx(119), bestx(120),
+ bestx(121), bestx(122), bestx(123), bestx(124), bestx(125),
+ bestx(126), bestx(127), bestx(128), bestx(129), bestx(130),
+ bestx(131), bestx(132), bestx(133), bestx(134), bestx(135),
+ bestx(136), bestx(137), bestx(138), bestx(139), bestx(140),
+ bestx(141), bestx(142), bestx(143), bestx(144), bestx(145),
+ bestx(146), bestx(147), bestx(148), bestx(149), bestx(150),
+ bestx(151), bestx(152), bestx(153), bestx(154), bestx(155),
+ bestx(156), bestx(157), bestx(158), bestx(159), bestx(160),
+ bestx(161), bestx(162), bestx(163), bestx(164), bestx(165),
+ bestx(166), bestx(167), bestx(168), bestx(169), bestx(170),
+ bestx(171), bestx(172), bestx(173), bestx(174), bestx(175),
+ bestx(176), bestx(177), bestx(178), bestx(179), bestx(180),
+ bestx(181), bestx(182), bestx(183), bestx(184), bestx(185),
+ bestx(186), bestx(187), bestx(188), bestx(189), bestx(190),
+ bestx(191), bestx(192), bestx(193), bestx(194), bestx(195),
+ bestx(196), bestx(197), bestx(198), bestx(199), bestx(200),
+ bestx(201), bestx(202), bestx(203), bestx(204), bestx(205),
+ bestx(206), bestx(207), bestx(208), bestx(209), bestx(210),
+ bestfval
       write (5, *) bestfval, iter, elapsed(1)-ltime
   if (bestfval .lt. bestfvalt) then
       bestfvalt = bestfval
       bestxt(1) = bestx(1)
       bestxt(2) = bestx(2)
       bestxt(3) = bestx(3)
       bestxt(4) = bestx(4)
       bestxt(5) = bestx(5)
       bestxt(6) = bestx(6)
       bestxt(7) = bestx(7)
       bestxt(8) = bestx(8)
       bestxt(9) = bestx(9)
       bestxt(10) = bestx(10)
       bestxt(11) = bestx(11)
       bestxt(12) = bestx(12)
       bestxt(13) = bestx(13)
       bestxt(14) = bestx(14)
       bestxt(15) = bestx(15)
       bestxt(16) = bestx(16)
       bestxt(17) = bestx(17)
       bestxt(18) = bestx(18)
       bestxt(19) = bestx(19)
       bestxt(20) = bestx(20)
       bestxt(21) = bestx(21)
       bestxt(22) = bestx(22)
       bestxt(23) = bestx(23)
       bestxt(24) = bestx(24)
       bestxt(25) = bestx(25)
       bestxt(26) = bestx(26)
       bestxt(27) = bestx(27)
       bestxt(28) = bestx(28)
       bestxt(29) = bestx(29)
       bestxt(30) = bestx(30)
       bestxt(31) = bestx(31)
       bestxt(32) = bestx(32)
       bestxt(33) = bestx(33)
       bestxt(34) = bestx(34)
       bestxt(35) = bestx(35)
       bestxt(36) = bestx(36)
       bestxt(37) = bestx(37)
       bestxt(38) = bestx(38)
       bestxt(39) = bestx(39)
       bestxt(40) = bestx(40)
       bestxt(41) = bestx(41)
       bestxt(42) = bestx(42)
       bestxt(43) = bestx(43)
       bestxt(44) = bestx(44)
       bestxt(45) = bestx(45)
```

```
bestxt(46) = bestx(46)
bestxt(47) = bestx(47)
bestxt(48) = bestx(48)
bestxt(49) = bestx(49)
bestxt(50) = bestx(50)
bestxt(51) = bestx(51)
bestxt(52) = bestx(52)
bestxt(53) = bestx(53)
bestxt(54) = bestx(54)
bestxt(55) = bestx(55)
bestxt(56) = bestx(56)
bestxt(57) = bestx(57)
bestxt(58) = bestx(58)
bestxt(59) = bestx(59)
bestxt(60) = bestx(60)
bestxt(61) = bestx(61)
bestxt(62) = bestx(62)
bestxt(63) = bestx(63)
bestxt(64) = bestx(64)
bestxt(65) = bestx(65)
bestxt(66) = bestx(66)
bestxt(67) = bestx(67)
bestxt(68) = bestx(68)
bestxt(69) = bestx(69)
bestxt(70) = bestx(70)
bestxt(71) = bestx(71)
bestxt(72) = bestx(72)
bestxt(73) = bestx(73)
bestxt(74) = bestx(74)
bestxt(75) = bestx(75)
bestxt(76) = bestx(76)
bestxt(77) = bestx(77)
bestxt(78) = bestx(78)
bestxt(79) = bestx(79)
bestxt(80) = bestx(80)
bestxt(81) = bestx(81)
bestxt(82) = bestx(82)
bestxt(83) = bestx(83)
bestxt(84) = bestx(84)
bestxt(85) = bestx(85)
bestxt(86) = bestx(86)
bestxt(87) = bestx(87)
bestxt(88) = bestx(88)
bestxt(89) = bestx(89)
bestxt(90) = bestx(90)
bestxt(91) = bestx(91)
bestxt(92) = bestx(92)
bestxt(93) = bestx(93)
bestxt(94) = bestx(94)
bestxt(95) = bestx(95)
bestxt(96) = bestx(96)
bestxt(97) = bestx(97)
bestxt(98) = bestx(98)
bestxt(99) = bestx(99)
bestxt(100) = bestx(100)
bestxt(101) = bestx(101)
bestxt(102) = bestx(102)
bestxt(103) = bestx(103)
bestxt(104) = bestx(104)
bestxt(105) = bestx(105)
bestxt(106) = bestx(106)
bestxt(107) = bestx(107)
bestxt(108) = bestx(108)
bestxt(109) = bestx(109)
bestxt(110) = bestx(110)
bestxt(111) = bestx(111)
bestxt(112) = bestx(112)
bestxt(113) = bestx(113)
bestxt(114) = bestx(114)
bestxt(115) = bestx(115)
bestxt(116) = bestx(116)
bestxt(117) = bestx(117)
bestxt(118) = bestx(118)
bestxt(119) = bestx(119)
bestxt(120) = bestx(120)
bestxt(121) = bestx(121)
bestxt(122) = bestx(122)
bestxt(123) = bestx(123)
bestxt(124) = bestx(124)
bestxt(125) = bestx(125)
```

```
bestxt(126) = bestx(126)
bestxt(127) = bestx(127)
bestxt(128) = bestx(128)
bestxt(129) = bestx(129)
bestxt(130) = bestx(130)
bestxt(131) = bestx(131)
bestxt(132) = bestx(132)
bestxt(133) = bestx(133)
bestxt(134) = bestx(134)
bestxt(135) = bestx(135)
bestxt(136) = bestx(136)
bestxt(137) = bestx(137)
bestxt(138) = bestx(138)
bestxt(139) = bestx(139)
bestxt(140) = bestx(140)
bestxt(141) = bestx(141)
bestxt(142) = bestx(142)
bestxt(143) = bestx(143)
bestxt(144) = bestx(144)
bestxt(145) = bestx(145)
bestxt(146) = bestx(146)
bestxt(147) = bestx(147)
bestxt(148) = bestx(148)
bestxt(149) = bestx(149)
bestxt(150) = bestx(150)
bestxt(151) = bestx(151)
bestxt(152) = bestx(152)
bestxt(153) = bestx(153)
bestxt(154) = bestx(154)
bestxt(155) = bestx(155)
bestxt(156) = bestx(156)
bestxt(157) = bestx(157)
bestxt(158) = bestx(158)
bestxt(159) = bestx(159)
bestxt(160) = bestx(160)
bestxt(161) = bestx(161)
bestxt(162) = bestx(162)
bestxt(163) = bestx(163)
bestxt(164) = bestx(164)
bestxt(165) = bestx(165)
bestxt(166) = bestx(166)
bestxt(167) = bestx(167)
bestxt(168) = bestx(168)
bestxt(169) = bestx(169)
bestxt(170) = bestx(170)
bestxt(171) = bestx(171)
bestxt(172) = bestx(172)
bestxt(173) = bestx(173)
bestxt(174) = bestx(174)
bestxt(175) = bestx(175)
bestxt(176) = bestx(176)
bestxt(177) = bestx(177)
bestxt(178) = bestx(178)
bestxt(179) = bestx(179)
bestxt(180) = bestx(180)
bestxt(181) = bestx(181)
bestxt(182) = bestx(182)
bestxt(183) = bestx(183)
bestxt(184) = bestx(184)
bestxt(185) = bestx(185)
bestxt(186) = bestx(186)
bestxt(187) = bestx(187)
bestxt(188) = bestx(188)
bestxt(189) = bestx(189)
bestxt(190) = bestx(190)
bestxt(191) = bestx(191)
bestxt(192) = bestx(192)
bestxt(193) = bestx(193)
bestxt(194) = bestx(194)
bestxt(195) = bestx(195)
bestxt(196) = bestx(196)
bestxt(197) = bestx(197)
bestxt(198) = bestx(198)
bestxt(199) = bestx(199)
bestxt(200) = bestx(200)
bestxt(201) = bestx(201)
bestxt(202) = bestx(202)
bestxt(203) = bestx(203)
bestxt(204) = bestx(204)
bestxt(205) = bestx(205)
```

```
      bestxt(206) = bestx(206)
      bestxt(207) = bestx(207)
      bestxt(208) = bestx(208)
      bestxt(209) = bestx(209)
      bestxt(210) = bestx(210)
    end if
    if (bestfval .gt. worstfvalt) then
        worstfvalt = bestfval
      worstxt(1)= bestx(1)
      worstxt(2)= bestx(2)
      worstxt(3)= bestx(3)
      worstxt(4)= bestx(4)
      worstxt(5)= bestx(5)
      worstxt(6)= bestx(6)
      worstxt(7)= bestx(7)
      worstxt(8)= bestx(8)
      worstxt(9)= bestx(9)
      worstxt(10)= bestx(10)
      worstxt(11)= bestx(11)
      worstxt(12)= bestx(12)
      worstxt(13)= bestx(13)
      worstxt(14)= bestx(14)
      worstxt(15)= bestx(15)
      worstxt(16)= bestx(16)
      worstxt(17)= bestx(17)
      worstxt(18)= bestx(18)
      worstxt(19)= bestx(19)
      worstxt(20)= bestx(20)
      worstxt(21)= bestx(21)
      worstxt(22)= bestx(22)
      worstxt(23)= bestx(23)
      worstxt(24)= bestx(24)
      worstxt(25)= bestx(25)
      worstxt(26)= bestx(26)
      worstxt(27)= bestx(27)
      worstxt(28)= bestx(28)
      worstxt(29)= bestx(29)
      worstxt(30)= bestx(30)
      worstxt(31)= bestx(31)
      worstxt(32)= bestx(32)
      worstxt(33)= bestx(33)
      worstxt(34)= bestx(34)
      worstxt(35)= bestx(35)
      worstxt(36)= bestx(36)
      worstxt(37)= bestx(37)
      worstxt(38)= bestx(38)
      worstxt(39)= bestx(39)
      worstxt(40)= bestx(40)
      worstxt(41)= bestx(41)
      worstxt(42)= bestx(42)
      worstxt(43)= bestx(43)
      worstxt(44)= bestx(44)
      worstxt(45)= bestx(45)
      worstxt(46)= bestx(46)
      worstxt(47)= bestx(47)
      worstxt(48)= bestx(48)
      worstxt(49)= bestx(49)
      worstxt(50)= bestx(50)
      worstxt(51)= bestx(51)
      worstxt(52)= bestx(52)
      worstxt(53)= bestx(53)
      worstxt(54)= bestx(54)
      worstxt(55)= bestx(55)
      worstxt(56)= bestx(56)
      worstxt(57)= bestx(57)
      worstxt(58)= bestx(58)
      worstxt(59)= bestx(59)
      worstxt(60)= bestx(60)
      worstxt(61)= bestx(61)
      worstxt(62)= bestx(62)
      worstxt(63)= bestx(63)
      worstxt(64)= bestx(64)
      worstxt(65)= bestx(65)
      worstxt(66)= bestx(66)
      worstxt(67)= bestx(67)
      worstxt(68)= bestx(68)
      worstxt(69)= bestx(69)
      worstxt(70)= bestx(70)
      worstxt(71)= bestx(71)
      worstxt(72)= bestx(72)
```

worstxt(73)= bestx(73)
worstxt(74)= bestx(74)
worstxt(75)= bestx(75)
worstxt(76)= bestx(76)
worstxt(77)= bestx(77)
worstxt(78)= bestx(78)
worstxt(79)= bestx(79)
worstxt(80)= bestx(80)
worstxt(81)= bestx(81)
worstxt(82)= bestx(82)
worstxt(83)= bestx(83)
worstxt(84)= bestx(84)
worstxt(85)= bestx(85)
worstxt(86)= bestx(86)
worstxt(87)= bestx(87)
worstxt(88)= bestx(88)
worstxt(89)= bestx(89)
worstxt(90)= bestx(90)
worstxt(91)= bestx(91)
worstxt(92)= bestx(92)
worstxt(93)= bestx(93)
worstxt(94)= bestx(94)
worstxt(95)= bestx(95)
worstxt(96)= bestx(96)
worstxt(97)= bestx(97)
worstxt(98)= bestx(98)
worstxt(99)= bestx(99)
worstxt(100)= bestx(100)
worstxt(101)= bestx(101)
worstxt(102)= bestx(102)
worstxt(103)= bestx(103)
worstxt(104)= bestx(104)
worstxt(105)= bestx(105)
worstxt(106)= bestx(106)
worstxt(107)= bestx(107)
worstxt(108)= bestx(108)
worstxt(109)= bestx(109)
worstxt(110)= bestx(110)
worstxt(111)= bestx(111)
worstxt(112)= bestx(112)
worstxt(113)= bestx(113)
worstxt(114)= bestx(114)
worstxt(115)= bestx(115)
worstxt(116)= bestx(116)
worstxt(117)= bestx(117)
worstxt(118)= bestx(118)
worstxt(119)= bestx(119)
worstxt(120)= bestx(120)
worstxt(121)= bestx(121)
worstxt(122)= bestx(122)
worstxt(123)= bestx(123)
worstxt(124)= bestx(124)
worstxt(125)= bestx(125)
worstxt(126)= bestx(126)
worstxt(127)= bestx(127)
worstxt(128)= bestx(128)
worstxt(129)= bestx(129)
worstxt(130)= bestx(130)
worstxt(131)= bestx(131)
worstxt(132)= bestx(132)
worstxt(133)= bestx(133)
worstxt(134)= bestx(134)
worstxt(135)= bestx(135)
worstxt(136)= bestx(136)
worstxt(137)= bestx(137)
worstxt(138)= bestx(138)
worstxt(139)= bestx(139)
worstxt(140)= bestx(140)
worstxt(141)= bestx(141)
worstxt(142)= bestx(142)
worstxt(143)= bestx(143)
worstxt(144)= bestx(144)
worstxt(145)= bestx(145)
worstxt(146)= bestx(146)
worstxt(147)= bestx(147)
worstxt(148)= bestx(148)
worstxt(149)= bestx(149)
worstxt(150)= bestx(150)
worstxt(151)= bestx(151)
worstxt(152)= bestx(152)

```
            worstxt(153)= bestx(153)
            worstxt(154)= bestx(154)
            worstxt(155)= bestx(155)
            worstxt(156)= bestx(156)
            worstxt(157)= bestx(157)
            worstxt(158)= bestx(158)
            worstxt(159)= bestx(159)
            worstxt(160)= bestx(160)
            worstxt(161)= bestx(161)
            worstxt(162)= bestx(162)
            worstxt(163)= bestx(163)
            worstxt(164)= bestx(164)
            worstxt(165)= bestx(165)
            worstxt(166)= bestx(166)
            worstxt(167)= bestx(167)
            worstxt(168)= bestx(168)
            worstxt(169)= bestx(169)
            worstxt(170)= bestx(170)
            worstxt(171)= bestx(171)
            worstxt(172)= bestx(172)
            worstxt(173)= bestx(173)
            worstxt(174)= bestx(174)
            worstxt(175)= bestx(175)
            worstxt(176)= bestx(176)
            worstxt(177)= bestx(177)
            worstxt(178)= bestx(178)
            worstxt(179)= bestx(179)
            worstxt(180)= bestx(180)
            worstxt(181)= bestx(181)
            worstxt(182)= bestx(182)
            worstxt(183)= bestx(183)
            worstxt(184)= bestx(184)
            worstxt(185)= bestx(185)
            worstxt(186)= bestx(186)
            worstxt(187)= bestx(187)
            worstxt(188)= bestx(188)
            worstxt(189)= bestx(189)
            worstxt(190)= bestx(190)
            worstxt(191)= bestx(191)
            worstxt(192)= bestx(192)
            worstxt(193)= bestx(193)
            worstxt(194)= bestx(194)
            worstxt(195)= bestx(195)
            worstxt(196)= bestx(196)
            worstxt(197)= bestx(197)
            worstxt(198)= bestx(198)
            worstxt(199)= bestx(199)
            worstxt(200)= bestx(200)
            worstxt(201)= bestx(201)
            worstxt(202)= bestx(202)
            worstxt(203)= bestx(203)
            worstxt(204)= bestx(204)
            worstxt(205)= bestx(205)
            worstxt(206)= bestx(206)
            worstxt(207)= bestx(207)
            worstxt(208)= bestx(208)
            worstxt(209)= bestx(209)
            worstxt(210)= bestx(210)
        end if

         fvaltave = fvaltave + bestfval

        if ((elapsed(1)-ltime) .lt. besttt)
     +      besttt = elapsed(1)-ltime
         if ((elapsed(1)-ltime) .gt. worsttt)
     +      worsttt = elapsed(1)-ltime

         ttave = ttave + (elapsed(1)-ltime)

        if (real(iter) .lt. bestit)
     +      bestit = real(iter)
         if (real(iter) .gt. worstit)
     +      worstit = real(iter)
         itave = itave + real(iter)

        write (4, *) '];'
 9911 continue
         write (3, *) '];'
        write (5, *) '];'
       write (3, *) 'best = [', bestxt(1), bestxt(2), bestxt(3),
```

```
+ bestxt(4), bestxt(5), bestxt(6), bestxt(7), bestxt(8), bestxt(9),
+ bestxt(10),
+ bestxt(11), bestxt(12), bestxt(13), bestxt(14), bestxt(15),
+ bestxt(16), bestxt(17), bestxt(18), bestxt(19), bestxt(20),
+ bestxt(21), bestxt(22), bestxt(23), bestxt(24), bestxt(25),
+ bestxt(26), bestxt(27), bestxt(28), bestxt(29), bestxt(30),
+ bestxt(31), bestxt(32), bestxt(33), bestxt(34), bestxt(35),
+ bestxt(36), bestxt(37), bestxt(38), bestxt(39), bestxt(40),
+ bestxt(41), bestxt(42), bestxt(43), bestxt(44), bestxt(45),
+ bestxt(46), bestxt(47), bestxt(48), bestxt(49), bestxt(50),
+ bestxt(51), bestxt(52), bestxt(53), bestxt(54), bestxt(55),
+ bestxt(56), bestxt(57), bestxt(58), bestxt(59), bestxt(60),
+ bestxt(61), bestxt(62), bestxt(63), bestxt(64), bestxt(65),
+ bestxt(66), bestxt(67), bestxt(68), bestxt(69), bestxt(70),
+ bestxt(71), bestxt(72), bestxt(73), bestxt(74), bestxt(75),
+ bestxt(76), bestxt(77), bestxt(78), bestxt(79), bestxt(80),
+ bestxt(81), bestxt(82), bestxt(83), bestxt(84), bestxt(85),
+ bestxt(86), bestxt(87), bestxt(88), bestxt(89), bestxt(90),
+ bestxt(91), bestxt(92), bestxt(93), bestxt(94), bestxt(95),
+ bestxt(96), bestxt(97), bestxt(98), bestxt(99), bestxt(100),
+ bestxt(101), bestxt(102), bestxt(103), bestxt(104), bestxt(105),
+ bestxt(106), bestxt(107), bestxt(108), bestxt(109), bestxt(110),
+ bestxt(111), bestxt(112), bestxt(113), bestxt(114), bestxt(115),
+ bestxt(116), bestxt(117), bestxt(118), bestxt(119), bestxt(120),
+ bestxt(121), bestxt(122), bestxt(123), bestxt(124), bestxt(125),
+ bestxt(126), bestxt(127), bestxt(128), bestxt(129), bestxt(130),
+ bestxt(131), bestxt(132), bestxt(133), bestxt(134), bestxt(135),
+ bestxt(136), bestxt(137), bestxt(138), bestxt(139), bestxt(140),
+ bestxt(141), bestxt(142), bestxt(143), bestxt(144), bestxt(145),
+ bestxt(146), bestxt(147), bestxt(148), bestxt(149), bestxt(150),
+ bestxt(151), bestxt(152), bestxt(153), bestxt(154), bestxt(155),
+ bestxt(156), bestxt(157), bestxt(158), bestxt(159), bestxt(160),
+ bestxt(161), bestxt(162), bestxt(163), bestxt(164), bestxt(165),
+ bestxt(166), bestxt(167), bestxt(168), bestxt(169), bestxt(170),
+ bestxt(171), bestxt(172), bestxt(173), bestxt(174), bestxt(175),
+ bestxt(176), bestxt(177), bestxt(178), bestxt(179), bestxt(180),
+ bestxt(181), bestxt(182), bestxt(183), bestxt(184), bestxt(185),
+ bestxt(186), bestxt(187), bestxt(188), bestxt(189), bestxt(190),
+ bestxt(191), bestxt(192), bestxt(193), bestxt(194), bestxt(195),
+ bestxt(196), bestxt(197), bestxt(198), bestxt(199), bestxt(200),
+ bestxt(201), bestxt(202), bestxt(203), bestxt(204), bestxt(205),
+ bestxt(206), bestxt(207), bestxt(208), bestxt(209), bestxt(210),
+       bestfvalt, bestit, besttt, '];'
    write (3, *) 'average = [', fvaltave/maxtrial,
+                   itave/maxtrial, ttave/maxtrial, '];'
   write (3, *) 'worst = [', worstxt(1), worstxt(2), worstxt(3),
+ worstxt(4), worstxt(5), worstxt(6), worstxt(7), worstxt(8),
+ worstxt(9), worstxt(10),
+ worstxt(11), worstxt(12), worstxt(13), worstxt(14), worstxt(15),
+ worstxt(16), worstxt(17), worstxt(18), worstxt(19), worstxt(20),
+ worstxt(21), worstxt(22), worstxt(23), worstxt(24), worstxt(25),
+ worstxt(26), worstxt(27), worstxt(28), worstxt(29), worstxt(30),
+ worstxt(31), worstxt(32), worstxt(33), worstxt(34), worstxt(35),
+ worstxt(36), worstxt(37), worstxt(38), worstxt(39), worstxt(40),
+ worstxt(41), worstxt(42), worstxt(43), worstxt(44), worstxt(45),
+ worstxt(46), worstxt(47), worstxt(48), worstxt(49), worstxt(50),
+ worstxt(51), worstxt(52), worstxt(53), worstxt(54), worstxt(55),
+ worstxt(56), worstxt(57), worstxt(58), worstxt(59), worstxt(60),
+ worstxt(61), worstxt(62), worstxt(63), worstxt(64), worstxt(65),
+ worstxt(66), worstxt(67), worstxt(68), worstxt(69), worstxt(70),
+ worstxt(71), worstxt(72), worstxt(73), worstxt(74), worstxt(75),
+ worstxt(76), worstxt(77), worstxt(78), worstxt(79), worstxt(80),
+ worstxt(81), worstxt(82), worstxt(83), worstxt(84), worstxt(85),
+ worstxt(86), worstxt(87), worstxt(88), worstxt(89), worstxt(90),
+ worstxt(91), worstxt(92), worstxt(93), worstxt(94), worstxt(95),
+ worstxt(96), worstxt(97), worstxt(98), worstxt(99), worstxt(100),
+ worstxt(101), worstxt(102), worstxt(103), worstxt(104),
+ worstxt(105), worstxt(106), worstxt(107), worstxt(108),
+ worstxt(109), worstxt(110), worstxt(111), worstxt(112),
+ worstxt(113), worstxt(114), worstxt(115), worstxt(116),
+ worstxt(117), worstxt(118), worstxt(119), worstxt(120),
+ worstxt(121), worstxt(122), worstxt(123), worstxt(124),
+ worstxt(125), worstxt(126), worstxt(127), worstxt(128),
+ worstxt(129), worstxt(130), worstxt(131), worstxt(132),
+ worstxt(133), worstxt(134), worstxt(135), worstxt(136),
+ worstxt(137), worstxt(138), worstxt(139), worstxt(140),
+ worstxt(141), worstxt(142), worstxt(143), worstxt(144),
+ worstxt(145), worstxt(146), worstxt(147), worstxt(148),
+ worstxt(149), worstxt(150), worstxt(151), worstxt(152),
+ worstxt(153), worstxt(154), worstxt(155), worstxt(156),
```

```
     + worstxt(157), worstxt(158), worstxt(159), worstxt(160),
     + worstxt(161), worstxt(162), worstxt(163), worstxt(164),
     + worstxt(165), worstxt(166), worstxt(167), worstxt(168),
     + worstxt(169), worstxt(170), worstxt(171), worstxt(172),
     + worstxt(173), worstxt(174), worstxt(175), worstxt(176),
     + worstxt(177), worstxt(178), worstxt(179), worstxt(180),
     + worstxt(181), worstxt(182), worstxt(183), worstxt(184),
     + worstxt(185), worstxt(186), worstxt(187), worstxt(188),
     + worstxt(189), worstxt(190), worstxt(191), worstxt(192),
     + worstxt(193), worstxt(194), worstxt(195), worstxt(196),
     + worstxt(197), worstxt(198), worstxt(199), worstxt(200),
     + worstxt(201), worstxt(202), worstxt(203), worstxt(204),
     + worstxt(205), worstxt(206), worstxt(207), worstxt(208),
     + worstxt(209), worstxt(210),
     +                   worstfvalt, worstit, worsttt, '];'
      close(3)
      close(4)
      close(5)

      stop
      end

      function heatrate(no, megawt, mwlow, mwhi)

      implicit none
      integer no
      real heatrate, megawt, mwlow, mwhi, gcost, ocost
      real trfactor, opfactor
      real coef_a, coef_b, coef_c

*     Gas & Oil (Baht/GCal)
      gcost = 500.25
      ocost = 602.73

*     Transform BTU->Cal
      trfactor = 251.996

      if (megawt .lt. mwlow) then
        heatrate = 1000000000.0
        goto 9010
      else if (megawt .gt. mwhi) then
        heatrate = 1000000000.0
        goto 9010
      end if

      if (no .eq. 1) then
*     KN-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.010
      goto 9110
      else if (no .eq. 2) then
*     RPB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.015
      goto 9110
      else if (no .eq. 3) then
*     RPB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 4) then
*     RPB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 5) then
*     MM-T8
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.010
      goto 9110
      else if (no .eq. 6) then
*     MM-T9
      coef_a = 0.0
      coef_b = 9147.0
      coef_c = 55117.015
```

```
      goto 9110
      else if (no .eq. 7) then
*     MM-T4
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.000
      goto 9110
      else if (no .eq. 8) then
*     MM-T5
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.010
      goto 9110
      else if (no .eq. 9) then
*     MM-T6
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.015
      goto 9110
      else if (no .eq. 10) then
*     MM-T7
      coef_a = 0.0
      coef_b = 10126.0
      coef_c = 58482.020
      goto 9110
      else if (no .eq. 11) then
*     SK-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.030
      goto 9110
      else if (no .eq. 12) then
*     SK-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.035
      goto 9110
      else if (no .eq. 13) then
*     SK-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.040
      goto 9110
      else if (no .eq. 14) then
*     BB-H8
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.045
      goto 9110
      else if (no .eq. 15) then
*     BB-H7
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.050
      goto 9110
      else if (no .eq. 16) then
*     BB-H6

      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.055
      goto 9110
      else if (no .eq. 17) then
*     BB-H4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.060
      goto 9110
      else if (no .eq. 18) then
*     BB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.065
      goto 9110
      else if (no .eq. 19) then
*     NPO-C1
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
```

```
      else if (no .eq. 20) then
*     IPT-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.070
      goto 9110
      else if (no .eq. 21) then
*     BPK-C1
      coef_a = 22.4
      coef_b = 5760.0
      coef_c = 222000.0
      goto 9110
      else if (no .eq. 22) then
*     BPK-T2
      coef_a = 1.56
      coef_b = 8734.0
      coef_c = 59400.0
      goto 9110
      else if (no .eq. 23) then
*     SB-C1
      coef_a = -0.68
      coef_b = 6653.0
      coef_c = 114037.0
      goto 9110
      else if (no .eq. 24) then
*     SB-T1
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.0
      goto 9110
      else if (no .eq. 25) then
*     SB-T2
      coef_a = 10.8
      coef_b = 7259.0
      coef_c = 114862.050
      goto 9110
      else if (no .eq. 26) then
*     SB-T3
      coef_a = 13.71
      coef_b = 6980.0
      coef_c = 141375.0
      goto 9110
      else if (no .eq. 27) then
*     KHL-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.075
      goto 9110
      else if (no .eq. 28) then
*     KHL-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.080
      goto 9110
      else if (no .eq. 29) then
*     KHL-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.085
      goto 9110
      else if (no .eq. 30) then
*     SNR-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.090
      goto 9110
      else if (no .eq. 31) then
*     SNR-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.095
      goto 9110
      else if (no .eq. 32) then
*     SNR-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.100
      goto 9110
      else if (no .eq. 33) then
*     SNR-H4
```

```
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.105
      goto 9110
      else if (no .eq. 34) then
*     TN-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.110
      goto 9110
      else if (no .eq. 35) then
*     TN-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.115
      goto 9110
      else if (no .eq. 36) then
*     RB-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.120
      goto 9110
      else if (no .eq. 37) then
*     RB-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.125
      goto 9110
      else if (no .eq. 38) then
*     THB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.130
      goto 9110
      else if (no .eq. 39) then
*     THB-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.135
      goto 9110
      else if (no .eq. 40) then
*     GCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.140
      goto 9110
      else if (no .eq. 41) then
*     HH-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.145
      goto 9110
      else if (no .eq. 42) then
*     HH-H2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.150
      goto 9110
      else if (no .eq. 43) then
*     AEP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.155
      goto 9110
      else if (no .eq. 44) then
*     TPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.160
      goto 9110
      else if (no .eq. 45) then
*     TTP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.165
      goto 9110
      else if (no .eq. 46) then
*     NPC-T1
      coef_a = 0.0
      coef_b = 0.01
```

```
      coef_c = 0.170
      goto 9110
      else if (no .eq. 47) then
*     IP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.175
      goto 9110
      else if (no .eq. 48) then
*     IP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.180
      goto 9110
      else if (no .eq. 49) then
*     BCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.185
      goto 9110
      else if (no .eq. 50) then
*     MTP-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.190
      goto 9110
      else if (no .eq. 51) then
*     MTP-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.195
      goto 9110
      else if (no .eq. 52) then
*     TCC-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.200
      goto 9110
      else if (no .eq. 53) then
*     TCC-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.205
      goto 9110
      else if (no .eq. 54) then
*     C0C0-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.210
      goto 9110
      else if (no .eq. 55) then
*     C0C0-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.215
      goto 9110
      else if (no .eq. 56) then
*     NPS-T1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.020
      goto 9110
      else if (no .eq. 57) then
*     NPS-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.025
      goto 9110
      else if (no .eq. 58) then
*     NPO-C2
      coef_a = 3.2
      coef_b = 6533.0
      coef_c = 118500.0
      goto 9110
      else if (no .eq. 59) then
*     WN-C3
      coef_a = -5.55
      coef_b = 7455.0
      coef_c = 50631.0
      goto 9110
```

```
      else if (no .eq. 60) then
*     KN-T2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.230
      goto 9110
      else if (no .eq. 61) then
*     RY-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.235
      goto 9110
      else if (no .eq. 62) then
*     RY-C4
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.240
      goto 9110
      else if (no .eq. 63) then
*     BB-H5
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.245
      goto 9110
      else if (no .eq. 64) then
*     BB-H3
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.250
      goto 9110
      else if (no .eq. 65) then
*     BB-H1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.255
      goto 9110
      else if (no .eq. 66) then
*     BLCP-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.260
      goto 9110
      else if (no .eq. 67) then
*     BLCP-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.265
      goto 9110
      else if (no .eq. 68) then
*     EPEC-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.270
      goto 9110
      else if (no .eq. 69) then
*     GULF-C1
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.275
      goto 9110
      else if (no .eq. 70) then
*     GULF-C2
      coef_a = 0.0
      coef_b = 0.01
      coef_c = 0.280
      goto 9110
      end if

 9110   opfactor = megawt/mwhi*100.0
       heatrate = gcost * trfactor *
     +    (coef_a*opfactor**2 + coef_b*opfactor + coef_c)
     +    / 1000000.0

 9010 return
      end

      subroutine chkmw(megawt, mwlow, mwhi)
      implicit none
      real megawt, mwlow, mwhi
      if (megawt .lt. mwlow) then
```

```
        megawt = mwlow
      else if (megawt .gt. mwhi) then
        megawt = mwhi
      end if
      return
      end

      subroutine ran3(idum,rand)
      implicit double precision (a-h,m,o-z)

      save
*     implicit real*4(m)
      parameter (mbig=4000000.,mseed=1618033.,mz=0.,fac=1./mbig)
*     parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1./mbig)
*
*     According to Knuth, any large mbig, and any smaller (but still large)
*     mseed can be substituted for the above values.
*
      dimension ma(55)
      data iff /0/
      if (idum.lt.0 .or. iff.eq.0) then
        iff=1
        mj=mseed-dble(iabs(idum))
        mj=dmod(mj,mbig)
        ma(55)=mj
        mk=1
        do 9010 i=1,54
           ii=mod(21*i,55)
           ma(ii)=mk
           mk=mj-mk
           if(mk.lt.mz) mk=mk+mbig
           mj=ma(ii)
 9010    continue
        do 9015 k=1,4
           do 9020 i=1,55
              ma(i)=ma(i)-ma(1+mod(i+30,55))
              if(ma(i).lt.mz) ma(i)=ma(i)+mbig
 9020       continue
 9015    continue
        inext=0
        inextp=31
        idum=1
      endif
      inext=inext+1
      if(inext.eq.56) inext=1
      inextp=inextp+1
      if(inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
      rand=mj*fac
      return
      end
```