INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-105

TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING

PART II - FORTRAN LISTING

J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu

September 1975

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

```
      SUBROUTINE TOLOPT (NR,KT,KR,KD,KP,NP,Z,I1,I2,AZ,AX,MU,NV,SAMPT,GRA
     1D,PL,PU,W1,CW,IB1,SG,I3,I4,X,EPS,G,PS,XB,IX,X1,X2,W,H,XE,INDX,GF,I
     2AA,IBB,A,T1,T1P,NSTEP,QSTEP,DISCR,XU,XL,ID,IB,ICHECK,IVAR,P1,P2,ES
     3TD,AL,GPHI,PHI)
      DIMENSION Z(1), I1(1), I2(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMPT
     1(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1), IB1(1), SG(1), I3(1),
     2I4(1)
      DIMENSION X(1), EPS(1), G(1), PS(1), XB(1), IX(1), X1(1), X2(1), W
     1(1), H(1), XE(NR,NP,1), INDX(1), GF(1)
      DIMENSION IAA(1), IBB(1), A(1), T1(1), T1P(1), PHI(1)
      DIMENSION NSTEP(1), QSTEP(1), DISCR(KD,1), XU(1), XL(1), ID(1), IB
     1(KD,1), ICHECK(1), IVAR(1), P1(1), P2(1), ESTD(1), AL(1)
      DIMENSION GPHI(KP,1)
      COMMON /DSPT03/ AM,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     10,IFN
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL1/ KRP1,NVC,NCC
      COMMON /TOL2/ IOP,MOP,IFC,FO
      COMMON /TOL5/ NCONS
      COMMON /TOL6/ IALF,NPC
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
      LOGICAL FO
C
C     IUPD=0,SELECTED MU-MATRIX SUPPLIED--THE PROGRAM GOES
C         DIRECTLY TO FINAL OPTIMIZATION
C     IUPD=1,SELECTED MU-MATRIX SUPPLIED--FURTHER UPDATING
C         PERFORMED BEFORE FINAL OPTIMIZATION
C     IUPD=2,TOLOPT SELECTS VERTICES--AFTER ONE SELECTION
C         EXIT IS CALLED SUCH THAT MANUAL ELIMINATION OF VER-
C         TICES BASED ON MAGNITUDE CONSIDERATIONS CAN BE
C         PERFORMED BY THE USER
C     IUPD=ANY OTHER VALUE--AUTOMATED VERTEX SELECTION
C
C     DEFAULT PARAMETERS USED
C     ND1-MAXIMUM NUMBER OF UPDATINGS PERFORMED,PARAMETER MOP
C     ND2-WEIGHTING FACTORS IN OBJECTIVE FUNCTION,VECTOR CW
C     ND3-ESTIMATED MINIMUM VALUES OF ARTIFICIAL UNCONSTRAINED
C         AND ACTUAL OBJECTIVE FUNCTION,EST AND EST1
C     ND4-PARAMETERS AO,AI,XMAL
C     ND5-PARAMETERS ZERO,ETA,INSOLN,BSOLN
C     ND6-PARAMETER P IN UPDATING PROCEDURE
C
      IOP=0
      IFC=0
      MOP=5
      IALF=0
      IF (ND1.NE.0) MOP=ND1
      IF (ND2.NE.0) GO TO 2
      DO 1 I=1,KT
      CW(I)=1.
    1 CONTINUE
    2 CONTINUE
      IF (ND3.NE.0) GO TO 3
      EST=0.
      EST1=0.
```

```
3     CONTINUE
      IF (ND4.NE.0) GO TO 4
      A0=10.
      AI=10.
      XMAL=1.E+5
4     CONTINUE
      IF (ND5.NE.0) GO TO 7
      B=ABS(SAMPT(2,1))
      IF (NSP.EQ.1) GO TO 6
      DO 5 I=2,NSP
      BB=ABS(SAMPT(2,I))
      B=AMIN1(B,BB)
5     CONTINUE
6     ZERC=-0.01*B
      ETA=1.E-4
      INSOLN=0
7     CONTINUE
      DO 9 I=1,KT
      IF (IDATA.NE.1) GO TO 8
      IKT=I+KT
      GPHI(I,1)=Z(I)
      GPHI(IKT,1)=Z(IKT)
8     IF (I1(I).EQ.3) GO TO 9
      Z(I)=SQRT(Z(I))
9     CONTINUE
      KRP1=KR+1
      CALL XZTRAN (Z,X,I1,I2,NR,KT,KD,KP,KR,NPC)
      IF (IUPD.EQ.0.OR.IUPD.EQ.1) GO TO 11
      NVC=NSP
      NCC=NVC+2*NPC
      NCONS=NCC+NEC
      DO 10 I=1,KT
      DO 10 J=1,NVC
      MU(I,J)=0
      NV(J)=1
10    CONTINUE
11    CONTINUE
      IF (IDATA.NE.1) GO TO 12
C
C     PRINTS INPUT DATA
C
      CALL DSPTI (KP,KT,NR,KD,Z,I1,W1,EPS,PS,NP,QSTEP,NSTEP,DISCR,SAMPT,
     1GPHI)
      IDATA=0
12    CONTINUE
      IF (IOPT2.NE.1) GO TO 13
C
C     GRADIENT CHECK
C
      CALL DSPTL (NSP,NEC,KT,KP,NR,W,Z,GRAD,SAMPT,PHI,GPHI,W1,I3)
      IOPT2=0
13    CONTINUE
      IF (IOPT6.NE.0) GO TO 14
      NCONS=0
      GO TO 35
14    CONTINUE
      IF (IUPD.EQ.0) GO TO 34
```

```
      NP1=NP
      PS1=PS(1)
      NP=1
      PS(1)=10.
      IF (ND6.NE.0) PS(1)=FLOAT(ND6)
      IOPTT=IOPT5
      IOPT6=2
      IF (IUPD.EQ.1) GO TO 16
      IOPTF=IOPT4
      IOPT4=-1
C
C     FEASIBILITY CHECK IS PERFORMED BEFORE FIRST CALL OF VERTST,
C         TO CHECK FEASIBILITY OF NOMINAL POINT
C
      IFC=IFC+1
      IF (IFC.EQ.1) GO TO 26
15    CONTINUE
      WRITE (6,43)
      IOP=IOP+1
      IF (ISCEME.NE.0) WRITE (6,46) IOP
      CALL VERTST (Z,KT,KR,SAMPT,NSP,ISCEME,MAXVN,MU,I3,NV,NVC,AX,SG,IB1
     1,GRAD,IWORST,NDIM,IUPD)
      IF (IUPD.EQ.2) CALL EXIT
      GO TO 17
16    IOP=1
      WRITE (6,43)
      NVC=NVSUM
17    CALL UPDATE (I3,I4,NV,NVCO,NVC,NSP,NN,KT,MU,IB1)
      NCC=NVC+2*NPC
      NCONS=NCC+NEC
      IF (IOP.NE.1) WRITE (6,47) NN,NCONS
      NSP1=NSP/7
      IF (NSP1.EQ.0) GO TO 19
      DO 18 I=1,NSP1
      II=7*I-6
      IK=II+6
      WRITE (6,48) (J,NV(J),J=II,IK)
18    CONTINUE
19    NSP2=NSP-(7*NSP1)
      IF (NSP2.EQ.0) GO TO 20
      II=7*NSP1+1
      WRITE (6,49) NSP2,(I,NV(I),I=II,NSP)
20    CONTINUE
      NX=1+(NVC/25)
      NX1=NX-1
      IF (NX.EQ.1) GO TO 23
      DO 22 K=1,NX1
      K1=(K-1)*25
      IF (K.EQ.1) WRITE (6,45)
      IF (K.NE.1) WRITE (6,44)
      DO 21 I=1,KT
      WRITE (6,50) (MU(I,J+K1),J=1,25)
21    CONTINUE
22    CONTINUE
      K1=NX1*25
23    IL=NVC-NX1*25
      IF (IL.EQ.0) GO TO 25
```

```
      IF (NX.EQ.1) K1=0
      IF (NX.EQ.1) WRITE (6,45)
      IF (NX.NE.1) WRITE (6,44)
      DO 24 I=1,KT
      WRITE (6,50) (MU(I,J+K1),J=1,IL)
24    CONTINUE
25    CONTINUE
      IF (NCONS.GT.NDIM) GO TO 36
      IF (NN.EQ.0) GO TO 32
      IOPT4=1000
      IFC=IFC+1
26    CALL DISOP2 (NR,X,EPS,G,NP,PS,KD,NSTEP,DISCR,QSTEP,XB,IX,X1,X2,XU,
     1XL,ID,IB,W,H,XE,ICHECK,IVAR,P2,P1,INDX,GPHI,IAA,IBB,GF,PHI,A,T1,T1
     2P,AL,ESTD,Z,I1,I2,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      IF (IFC.EQ.1.AND.IFLAGA.NE.0) RETURN
      IF (ALPHA.GT.XMAL) RETURN
      IF (IFC.EQ.1.OR.IALF.EQ.0) GO TO 27
      GO TO 29
27    DO 28 I=1,NR
      XB(I)=X(I)
28    CONTINUE
      IF (IALF.EQ.0) GO TO 31
      GO TO 15
29    CONTINUE
      DO 30 I=1,NR
      X(I)=XB(I)
30    CONTINUE
      IALF=0
      AO=ALPHA
31    CONTINUE
      IF (IOP.LT.MOP) GO TO 15
      GO TO 33
32    CONTINUE
      IOP=MOP
      WRITE (6,51)
33    CONTINUE
      IOP=IOP+1
      NP=NP1
      PS(1)=PS1
      IOPT6=IOPTT
      IOPT4=IOPTF
      GO TO 35
34    IOP=NP+1
      MOP=NP
      NVC=NVSUM
      NCC=NVC+2*NPC
      NCONS=NVSUM+2*NPC+NEC
35    FO=.T.
      CALL DISOP2 (NR,X,EPS,G,NP,PS,KD,NSTEP,DISCR,QSTEP,XB,IX,X1,X2,XU,
     1XL,ID,IB,W,H,XE,ICHECK,IVAR,P2,P1,INDX,GPHI,IAA,IBB,GF,PHI,A,T1,T1
     2P,AL,ESTD,Z,I1,I2,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      GO TO 37
36    WRITE (6,52)
      CALL EXIT
37    CONTINUE
      IF (MAXNOD.EQ.0) GO TO 40
      WRITE (6,43)
```

```
      DO 38 I=1,NR
      X(I)=XB(I)
38    CONTINUE
      DO 39 I=1,KP
      J=I2(I)
      IF (I1(I).EQ.3) GO TO 39
      Z(I)=X(J)
39    CONTINUE
      WRITE (6,53)
      WRITE (6,54)  (I,Z(I),I=1,KP)
      RETURN
40    CONTINUE
      DO 42 I=1,KP
      J=I2(I)
      IF (I1(I).EQ.3) GO TO 41
      Z(I)=X(J)
41    IF (I.LE.KT.AND.I1(I).NE.3)  Z(I)=Z(I)**2
42    CONTINUE
      RETURN
C
C
C
43    FORMAT (1H1)
44    FORMAT (//)
45    FORMAT (/)
46    FORMAT (1H0,10X,*DATA FROM VERTICES SELECTION NO*,I3,/,11X,34(*-*)
     1,/)
47    FORMAT (1X,I3,55H MORE VERTICES HAVE BEEN DETECTED IN UPDATING PRO
     1CEDURE,/1H0,49HTHE CURRENT NUMBER OF CONSTRAINTS IN PROBLEM ARE ,I
     23,/)
48    FORMAT (7(4H NV(,I2,2H)=,I2),/)
49    FORMAT (=(4H NV(,I2,2H)=,I2),/)
50    FORMAT (1X,/,25I3)
51    FORMAT (1H0,*MU-MATRIX HAS REMAINED UNCHANGED IN TWO CONSECUTIVE C
     1ALLS FOR VERTST*/1H0,*THE PROGRAM GOES ON WITH THE FINAL OPTIMIZAT
     2ION*,/)
52    FORMAT (1H0,*THE CHOSEN NUMBER OF CONSTRAINTS ARE LARGER THAN THE
     1SPECIFIED DIMENSION OF THE GPHI-MATRIX*/1H0,*YOUR PROGRAM HAS BEEN
     2 TERMINATED*/1H0,*PLEASE,RESTART WITH LARGER MATRICES AND VALUE ND
     3IM*,/)
53    FORMAT (1H0,12X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,13X,33(*-*),
     1/)
54    FORMAT (8X,*Z(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE VERTST (X,KT,KR,SAMPT,NSP,ISCEME,MAXVN,MU,IU,NV,NVC,AX,
     1SG,IB,GRAD,IWORST,NDIM,IUPD)
      DIMENSION X(1), AX(1), SAMPT(3,1), MU(KT,1), IB(1), GRAD(1), NV(1)
     1, SG(1), IU(1)
C
C     X IS THE 2*KT VECTOR CONTAINING TOLERANCES AND NOMINAL VALUES
C        IN THE SAME ORDER AS THE CIRCUIT
C     NSP IS THE NUMBER OF SAMPLE POINTS CHOSEN BY THE USER
C     SAMPT IS A 3 BY NSP MATRIX CONTAINING SAMPLE POINTS,
C        SPECIFICATIONS AND THE CORRESPONDING WEIGHTING
C        FACTORS TO DENOTE UPPER AND LOWER SPECIFICATIONS
C     MAXVN IS THE MAXIMUM ALLOWABLE NUMBER OF VERTICES PER
C        SAMPLE POINT
C     ISCEME=0        FOR FIRST SCHEME(SPECIAL CASE OF SECOND SCHEME)
C     ISCEME=+1 OR -1 FOR SECOND SCHEME
C     ISCEME=+2 OR -2 FOR THE THIRD SCHEME
C        WHERE A MINUS SIGN IN FRONT OF THE SCHEME-INDICATOR
C        INDICATES A FURTHER REDUCTION BASED ON
C        MAGNITUDE CONSIDERATIONS
C     IWORST=1,PRINT OUT EVERY VERTEX USING THE THIRD SCHEME
C     IWORST=2,PRINT OUT ONLY THOSE ASSOCIATED WITH
C        NEGATIVE CONSTRAINTS USING THE THIRD SCHEME
C     IUPD=2,PRINT OUT ALL DETECTED VERTICES AND CORRES-
C        PONDING CONSTRAINT VALUES USING THE SECOND AND THIRD SCHEME
C     OUTPUT FROM SUBROUTINE VERTST CONSISTS OF MU,IU,NV AND NVC
C     NV IS A VECTOR OF 2*NSP ELEMENTS WHICH INDICATES THE
C        NUMBER OF VERTICES CHOSEN AT EACH FREQUENCY POINT
C     NVC IS THE SUM OF ALL VERTICES
C     IU IS A VECTOR OF NVC ELEMENTS OF THE CHOSEN VERTICES
C     MU IS A MATRIX OF KT BY NVC ELEMENTS CORRESPONDING TO THE
C        BINARY VALUE OF R
C     AX,IB,SG,GRAD ARE WORKING ARRAYS
C     KT IS THE NUMBER OF TOLERANCED COMPONENTS
C
      NVC=0
      I=ISCEME+3
      GO TO (23,5,1,5,23), I
C
C     SCHEME 0--VERTICES CHOSEN BASED ON FIRST
C        DERIVATIVES EVALUATED AT THE NOMINAL POINT
C
1     CONTINUE
      DO 2 J=1,KT
      AX(J)=X(J+KT)
2     CONTINUE
      DO 4 IF=1,NSP
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,1)
      DO 3 I=1,KT
      MU(I,IF)=+SIGN(1.,SAMPT(3,IF)*GRAD(I))
      IB(I)=MU(I,IF)
3     CONTINUE
      NVC=NVC+1
      NV(IF)=1
      CALL BDDB (IB,J,KT,1)
      IU(NVC)=J
4     CONTINUE
      RETURN
```

```
    5 CONTINUE
C
C     SCHEME 2--BASED ON PERTUBATION OF ONE COMPONENT AT A TIME
C
      IPR=0
      DO 22 IF=1,NSP
      DO 6 J=1,KT
      AX(J)=X(J+KT)
    6 CONTINUE
      I1=1
      I2=1
      DO 10 I=1,KT
      IKT=I+KT
C
C     RELATIVE TOLERANCES
C
      IF (I.LE.KR) AX(I)=X(IKT)*(1.-X(I))
C
C     ABSOLUTE TOLERANCES
C
      IF (I.GT.KR) AX(I)=X(IKT)-X(I)
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,1)
      IM=+SIGN(1.,SAMPT(3,IF)*GRAD(I))
C
C     RELATIVE TOLERANCES
C
      IF (I.LE.KR) AX(I)=X(IKT)*(1.+X(I))
C
C     ABSOLUTE TOLERANCES
C
      IF (I.GT.KR) AX(I)=X(IKT)+X(I)
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,1)
      IP=+SIGN(1.,SAMPT(3,IF)*GRAD(I))
      AX(I)=X(IKT)
C
C     FILL UP THE ITH ROW OF THE MATRIX MU BY THE NEGATIVE
C        SIGNS OF THE PARTIAL DERIVATIVES
C
      DO 7 J=1,I1
      MU(I,J+NVC)=IM
    7 CONTINUE
      IF (IM.EQ.IP) GO TO 10
C
C     IF SIGN OF DERIVATIVES AT THE PERTURBED ENDS ARE
C        DIFFERENT ,THEN DOUBLE THE NUMBER OF COLUMNS OF MU,
C        FILL THE NUMBER OF COLUMNS BY COPYING THE FIRST
C        PART INTO THE SECONC ,EXCEPT FOR ROW NUMBER I.
C        THE FIRST HALF OF ITH ROW ARE GIVEN  VALUES OF IM,
C        THE SECOND HALF VALUES OF IP.
C
      IPR=IPR+1
      IF (IPR.EQ.1) WRITE (6,36)
      I1P=I1+1
      I1=2*I1
      IM1=I-1
      DO 9 J=I1P,I1
      JS=J+NVC
```

```
      IF (JS.GE.NDIM) GO TO 31
      DO 8 J1=1,IM1
      IF (IM1.EQ.0) GO TO 8
      MU(J1,JS)=MU(J1,JS-I2)
8     CONTINUE
      MU(I,JS)=IP
9     CONTINUE
      I2=I1
      IF (IM.LT.0.AND.IUPD.NE.2) WRITE (6,32) IF,SAMPT(1,IF)
10    CONTINUE
      NV(IF)=I2
      IF (IUPD.EQ.2) GO TO 13
      DO 12 I=1,I2
      DO 11 J=1,KT
      IB(J)=MU(J,I+NVC)
11    CONTINUE
      CALL BDDB (IB,I3,KT,1)
      IU(I+NVC)=I3
12    CONTINUE
      GO TO 19
13    CONTINUE
C
C     THIS PART PRINTS OUT DETECTED VERTICES FOR FURTHER
C        MANUAL ELIMINATION BASED ON MAGNITUDE CONSIDERATIONS
C
      DO 15 I=1,I2
      DO 14 J=1,KT
      IB(J)=MU(J,I)
14    CONTINUE
      CALL BDDB (IB,I3,KT,1)
      IU(I)=I3
15    CONTINUE
      DO 17 J=1,I2
      DO 16 I=1,KT
      IKT=I+KT
C
C     RELATIVE TOLERANCES
C
      IF (I.LE.KR) AX(I)=X(IKT)*(1.+MU(I,J)*X(I))
C
C     ABSOLUTE TOLERANCES
C
      IF (I.GT.KR) AX(I)=X(IKT)+MU(I,J)*X(I)
16    CONTINUE
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,0)
      SG(J)=SAMPT(3,IF)*(SAMPT(2,IF)-F)
17    CONTINUE
      WRITE (6,39) IF,SAMPT(1,IF)
      DO 18 J=1,I2
      WRITE (6,40) IU(J),SG(J)
18    CONTINUE
      GO TO 22
19    CONTINUE
      NVC1=NVC+1
      NVC=NVC+I2
C
C     FOLLOWING PART WILL ELIMINATE CERTAIN VERTICES BASED
```

```
C        ON CONSTRAINT VALUE CONSIDERATIONS
C
      IF (ISCEME.GT.0.OR.I2.LT.MAXVN) GO TO 22
      DO 21 J=NVC1,NVC
      DO 20 I=1,KT
      IKT=I+KT
C
C     RELATIVE TOLERANCES
C
      IF (I.LE.KR) AX(I)=X(IKT)*(1.+MU(I,J)*X(I))
C
C     ABSOLUTE TOLERANCES
C
      IF (I.GT.KR) AX(I)=X(IKT)+MU(I,J)*X(I)
   20 CONTINUE
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,0)
      SG(J)=SAMPT(3,IF)*(SAMPT(2,IF)-F)
   21 CONTINUE
      CALL SORT (SG,IU,I2,NVC)
      NVC=NVC+MAXVN-I2
      NV(IF)=MAXVN
   22 CONTINUE
      PRINT 38
      IF (IUPD.NE.2) WRITE (6,37)
      RETURN
   23 CONTINUE
C
C     SCHEME 3--ALL VERTICES ARE INVOLVED
C
      I1=2**KT
      DO 30 IF=1,NSP
      I2=0
      DO 29 I=1,I1
      CALL BDDB (IB,I,KT,0)
      DO 24 J=1,KT
      JK=J+KT
      IF (J.LE.KR) AX(J)=X(JK)*(1.+IB(J)*X(J))
      IF (J.GT.KR) AX(J)=X(JK)+IB(J)*X(J)
   24 CONTINUE
      CALL NETWRK (AX,SAMPT(1,IF),F,GRAD,1)
      G=SAMPT(3,IF)*(SAMPT(2,IF)-F)
      IF (I.EQ.1.AND.IUPD.EQ.2) WRITE (6,34) IF,SAMPT(1,IF)
      IF (IUPD.EQ.2) GO TO 28
      IF (IWORST.EQ.1) WRITE (6,33) IF,SAMPT(1,IF),I,G
      IF (IWORST.EQ.2.AND.G.LT.0.) WRITE (6,33) IF,SAMPT(1,IF),I,G
      DO 25 J=1,KT
      IB(J)=+SIGN(1.,SAMPT(3,IF)*GRAD(J))
   25 CONTINUE
      CALL BDDB (IB,IP,KT,1)
      IF (I.EQ.1) GO TO 27
      DO 26 J=1,I2
      IM=NVC+J
      IF (IM.GE.NDIM) GO TO 31
      IF (IP.EQ.IU(IM)) GO TO 29
   26 CONTINUE
   27 CONTINUE
      I2=I2+1
```

```
      IM=NVC+I2
      SG(IM)=G
      IU(IM)=IP
      GO TO 29
28    WRITE (6,35) I,G
29    CONTINUE
      IF (IUPD.EQ.2) GO TO 30
      NV(IF)=I2
      NVC=NVC+I2
      IF (NV(IF).GE.MAXVN) WRITE (6,42) IF,SAMPT(1,IF)
      IF (I2.LE.MAXVN.OR.ISCEME.GT.0) GO TO 30
      IF (NV(IF).EQ.1) GO TO 30
      CALL SORT (SG,IU,I2,NVC)
      NVC=NVC-I2+MAXVN
      NV(IF)=MAXVN
30    CONTINUE
      WRITE (6,38)
      IF (IUPD.NE.2) WRITE (6,37)
      RETURN
31    WRITE (6,41)
      CALL EXIT
      RETURN
C
C
C
32    FORMAT (8X,12H AT SAMPT(1,,I2,3H) =,F7.2)
33    FORMAT (8X,9H SAMPT(1,,I2,3H) =,F7.2,2X,9HVERTEX NO,I4,2X,2HG=,E13
     1.6)
34    FORMAT (1H ,*AT SAMPT(1,*,I2,*) =*,F7.2,1X,*ALL VERTICES HAVE BEEN
     1 CHECKED*//1H ,10X,*VERTEX NO*,10X,*CONSTRAINT VALUE*,/)
35    FORMAT (12X,I4,15X,E14.6)
36    FORMAT (/,68H AT FOLLOWING SAMPLE POINTS CONCAVITY VIOLATIONS HAVE
     1 BEEN INDICATED,/)
37    FORMAT (47H MU-MATRIX TO BE USED IN FOLLOWING OPTIMIZATION,/)
38    FORMAT (/)
39    FORMAT (1H ,*AT SAMPT(1,*,I2,*) =*,F7.2,1X,*FOLLOWING VERTICES HAV
     1E BEEN DETECTED*//1H ,10X,*VERTEX NO*,10X,*CONSTRAINT VALUE*,/)
40    FORMAT (12X,I4,15X,E14.6)
41    FORMAT (1H1,*THE NUMBER OF DETECTED VERTICES WILL FILL UP YOUR MU-
     1MATRIX*//1H ,*YOUR PROGRAM HAS BEEN TERMINATED*//1H ,*PLEASE RESTA
     2RT WITH LARGER MATRICES AND VALUE NDIM*,/)
42    FORMAT (* AT SAMPT(1,*,I2,*)=*,F7.2,1X,*THE NUMBER OF CHOSEN*/1H,*
     1VERTICES ARE LARGER THAN OR EQUAL TO MAXVN*,/)
      END
```

```
      SUBROUTINE UPDATE (IU,IUU,NV,NVCO,NVC,NSP,IUPD,KT,MU,IB)
      DIMENSION NV(1), MU(KT,1), IB(1), IU(1), IUU(1)
      LOGICAL FO
      COMMON /TOL2/ IOP,MOP,IFC,FO
C
C     THIS SUBROUTINE IS UPDATING THE CURRENT SET OF DETEC-
C        TED VERTICES.NEW ONES ARE STORED FOLLOWING THE
C        PREVIOUS DETECTED FOR EACH SAMPLE POINT
C
      IF (IOP.NE.1) GO TO 3
      DO 1 I=1,NVC
      IUU(I)=IU(I)
1     CONTINUE
      DO 2 I=1,NSP
      NV(I+NSP)=NV(I)
2     CONTINUE
      NVCO=NVC
      GO TO 13
3     KL=1
      KLO=1
      K=NV(1)
      KO=NV(1+NSP)
      IUPD=0
      DO 12 IF=1,NSP
      IT=0
      NS=0
      KO1=KO
      IF1=IF+1
      IF (IF.EQ.NSP) GO TO 5
      DO 4 I=IF1,NSP
      NS=NS+NV(I+NSP)
4     CONTINUE
      GO TO 6
5     NS=NV(NSP+NSP)
6     DO 10 J=KL,K
      DO 7 I=KLO,KO
      IF (IUU(I).EQ.IU(J)) GO TO 10
7     CONTINUE
      NVCO=NVCO+1
      IT=IT+1
      IUPD=IUPD+1
      IF (IF.EQ.NSP) GO TO 9
      N1=NVCO+1
      DO 8 JJ=1,NS
      IUU(N1-JJ)=IUU(NVCO-JJ)
8     CONTINUE
9     CONTINUE
      IUU(KO1+1)=IU(J)
      KO1=KO1+1
10    CONTINUE
      IF (IF.EQ.NSP) GO TO 11
      KL=KL+NV(IF)
      K=K+NV(IF+1)
      KLO=KLO+NV(IF+NSP)+IT
      KO=KO+NV(IF+NSP+1)+IT
11    NV(IF)=NV(IF+NSP)+IT
      NV(IF+NSP)=NV(IF)
```

```
12    CONTINUE
      NVC=NVCO
13    CONTINUE
      DO 14 KM=1,NVC
      CALL BDD3 (IB,IUU(KM),KT,0)
      DO 14 JM=1,KT
      MU(JM,KM)=IB(JM)
14    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONSTR (Z,X,I2,I5,AZ,AX,MU,GT,GGT,NV,SAMPT,GRSP,PL,PU,W
     1,NR,KT,KP)
      DIMENSION X(1), Z(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), GT(1),
     1 GGT(KP,1), NV(1), SAMPT(3,1), GRSP(1), PL(1), PU(1), W(1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL1/ KRP1,NVC,NCC
      COMMON /TOL5/ NCONS
C
C     REARRANGEMENT OF X-VECTOR IN USER SUPPLIED ORDER
C
      KR=KRP1-1
      DO 2 I=1,KP
      J=I5(I)
      IF (I2(I).EQ.3) GO TO 1
      Z(I)=X(J)
    1 IF (KK.EQ.0.AND.I.LE.KT) AZ(I)=Z(I)**2
      IF (KK.NE.0.AND.I.LE.KT) AZ(I)=Z(I)
    2 CONTINUE
      DO 4 I=1,KP
      DO 3 J=1,NCONS
      GGT(I,J)=0.
    3 CONTINUE
    4 CONTINUE
C
C     CALL IN EXTRA USER CONSTRAINTS
C
      IF (NEC.EQ.0) GO TO 7
      CALL USERCN (Z,GT,GGT,NR,KP)
      DO 5 I=1,NEC
      K3=I+NCC
      GT(K3)=GT(I)
      GT(I)=0.
    5 CONTINUE
      DO 6 J=1,NEC
      K3=J+NCC
      DO 6 I=1,KP
      GGT(I,K3)=GGT(I,J)
      GGT(I,J)=0.
    6 CONTINUE
    7 CONTINUE
      IDM=1
      IDX=0
      INDX=1
      DO 15 J=1,NVC
      L=0
      M=KT
      IDX=IDX+1
      IF (IDX.LE.NV(INDX)) GO TO 8
      IDX=1
      IDM=IDM+1
      INDX=INDX+1
    8 CONTINUE
      DO 9 I=1,KT
      K3=I+KT
```

```
      IF (I.LE.KR) AX(I)=Z(K3)*(1.+MU(I,J)*AZ(I))
      IF (I.GT.KR) AX(I)=Z(K3)+MU(I,J)*AZ(I)
9     CONTINUE
      CALL NETWRK (AX,SAMPT(1,IDM),RSP,GRSP,1)
C
C     CONSTRAINT FUNCTIONS
C
      GT(J)=SAMPT(3,IDM)*(SAMPT(2,IDM)-RSP)
      IF (KR.EQ.0) GO TO 12
      DO 11 I=1,KR
      K3=I+KT
      L=L+1
      IF (I2(I).EQ.3) GO TO 10
C
C     GRADIENTS WRT RELATIVE TOLERANCES
C
      IF (KK.EQ.0) GGT(L,J)=-2.*Z(K3)*MU(I,J)*Z(I)*GRSP(I)*SAMPT(3,IDM)
      IF (KK.NE.0) GGT(L,J)=-Z(K3)*MU(I,J)*GRSP(I)*SAMPT(3,IDM)
10    CONTINUE
      M=M+1
      IF (I2(K3).EQ.3) GO TO 11
C
C     GRADIENTS WRT NOMINAL VALUES
C
      GGT(M,J)=-(1.+MU(I,J)*AZ(I))*GRSP(I)*SAMPT(3,IDM)
11    CONTINUE
12    CONTINUE
      IF (KR.EQ.KT) GO TO 15
      DO 14 I=KRP1,KT
      K3=I+KT
      L=L+1
      IF (I2(I).EQ.3) GO TO 13
C
C     GRADIENTS WRT ABSOLUTE TOLERANCES
C
      IF (KK.EQ.0) GGT(L,J)=-2.*MU(I,J)*Z(I)*GRSP(I)*SAMPT(3,IDM)
      IF (KK.NE.0) GGT(L,J)=-MU(I,J)*GRSP(I)*SAMPT(3,IDM)
13    CONTINUE
      M=M+1
      IF (I2(K3).EQ.3) GO TO 14
C
C     GRADIENTS WRT NOMINAL VALUES
C
      GGT(M,J)=-GRSP(I)*SAMPT(3,IDM)
14    CONTINUE
15    CONTINUE
C
C     PARAMETER BOUND CONSTRAINTS
C
      J=NVC-1
      L=0
      M=KT
      IF (KR.EQ.0) GO TO 18
      DO 17 I=1,KR
      K3=I+KT
      IF (I2(I).EQ.3.AND.I2(K3).EQ.3) GO TO 17
      J=J+2
```

```
C
C      LOWER BOUND CONSTRAINTS FOR RELATIVE COMPONENTS
C
      GT(J)=Z(K3)*(1.-AZ(I))-PL(I)
      JJ=J+1
C
C      UPPER BOUND CONSTRAINTS FOR RELATIVE COMPONENTS
C
      GT(JJ)=PU(I)-Z(K3)*(1.+AZ(I))
      L=L+1
      IF (I2(I).EQ.3) GO TO 16
C
C     GRADIENTS WRT RELATIVE TOLERANCES
C
      IF (KK.EQ.0) GGT(L,J)=-2.*Z(K3)*Z(I)
      IF (KK.NE.0) GGT(L,J)=-Z(K3)
      GGT(L,JJ)=GGT(L,J)
16    CONTINUE
      M=M+1
      IF (I2(K3).EQ.3) GO TO 17
C
C     GRADIENTS WRT NOMINAL VALUES
C
      GGT(M,J)=1.-AZ(I)
      GGT(M,JJ)=-(1.+AZ(I))
17    CONTINUE
18    CONTINUE
      IF (KR.EQ.KT) GO TO 21
      DO 20 I=KRP1,KT
      K3=I+KT
      IF (I2(I).EQ.3.AND.I2(K3).EQ.3) GO TO 20
      J=J+2
C
C      LOWER BOUND CONSTRAINTS FOR ABSOLUTE COMPONENTS
C
      GT(J)=Z(K3)-AZ(I)-PL(I)
      JJ=J+1
C
C     UPPER BOUND CONSTRAINTS FOR ABSOLUTE COMPONENTS
C
      GT(JJ)=PU(I)-Z(K3)-AZ(I)
      L=L+1
      IF (I2(I).EQ.3) GO TO 19
C
C      GRADIENTS WRT ABSOLUTE TOLERANCES
C
      IF (KK.EQ.0) GGT(L,J)=-2.*Z(I)
      IF (KK.NE.0) GGT(L,J)=-1.
      GGT(L,JJ)=GGT(L,J)
19    CONTINUE
      M=M+1
      IF (I2(K3).EQ.3) GO TO 20
C
C      GRADIENTS WRT NOMINAL VALUES
C
      GGT(M,J)=1.
      GGT(M,JJ)=-1.
```

```
20    CONTINUE
21    CONTINUE
C
C     REARRANGEMENT OF GRADIENTS IN OPTIMIZATION ORDER
C
      DO 24 I=1,NCONS
      DO 22 J=1,KP
      M=I5(J)
      W(M)=GGT(J,I)
22    CONTINUE
      DO 23 J=1,NR
      GGT(J,I)=W(J)
23    CONTINUE
24    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XZTRAN (Z,X,I2,I5,NR,KT,KD,KP,KR,NPC)
      DIMENSION X(1), Z(1), I2(1), I5(1)
      LOGICAL FO
C
C     NR    NUMBER OF OPTIMIZATION VARIABLES
C     KD    NUMBER OF VARIABLES OF DISCRETE VALUE
C     KT    NUMBER OF TOLERANCED COMPONENTS
C     KDT   NUMBER OF DISCRETE TOLERANCED COMPONENTS
C     KRT   NUMBER OF DISCRETE TOLERANCED COMPONENTS OF GIVEN
C           RELATIVE VALUE
C     NPC   NUMBER OF COMPONENTS WITH BOUND CONSTRAINTS
C     Z     USER SUPPLIED VECTOR OF RELATIVE TOLERANCES,
C           ABSOLUTE TOLERANCES AND CORRESPONDING NOMINAL VALUES
C     I2    USER SUPPLIED VECTOR OF INDICATORS
C         1 FOR DISCRETE VALUE
C         2 FOR CONTINUOUS VALUE
C         3 FOR FIXED VALUE
C     I5    VECTOR OF REARRANGED INDICATORS
C
      COMMON /TOL2/ IOP,NOP,IFC,FO
      COMMON /DISTRA/ KDT,KRD
      J=0
      K=KD
      L=NR
      DO 4 I=1,KP
      M=I2(I)
      GO TO (1,2,3), M
1     J=J+1
      X(J)=Z(I)
      I5(I)=J
      GO TO 4
2     K=K+1
      X(K)=Z(I)
      I5(I)=K
      GO TO 4
3     L=L+1
      X(L)=Z(I)
      I5(I)=L
4     CONTINUE
      IF (IOP.GT.0) RETURN
      KDT=0
      KRD=0
      NPC=KT
      DO 5 I=1,KT
      K3=I+KT
      IF (I2(I).EQ.1) KDT=KDT+1
      IF (I2(I).EQ.1.AND.I.LE.KR) KRD=KRD+1
      IF (I2(I).EQ.3.AND.I2(K3).EQ.3) NPC=NPC-1
5     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BODB (IB,ID,N,IMODE)
C
C     N IS THE NUMBER OF BINARY DIGITS
C     IMODE=-1  CHANGE SIGN OF THE BINARY DIGITS
C     IMODE=0   CONVERT DECIMAL TO BINARY IN +1 AND -1 STATES
C     IMODE=+1  CONVERT -1 AND +1 BINARY DIGITS TO DECIMAL NUMBER
C
      DIMENSION IB(1)
      IF (IMODE) 7,4,1
1     DO 2 I=1,N
      IF (IB(I).LT.0) IB(I)=0
2     CONTINUE
      ID=IB(1)+1
      DO 3 I=2,N
      ID=ID+IB(I)*2**(I-1)
3     CONTINUE
      RETURN
4     ISUM=1
      DO 5 I=1,N
      J=N+1-I
      IB(J)=(ID-ISUM)/(2**(J-1))
      ISUM=ISUM+IB(J)*2**(J-1)
5     CONTINUE
      DO 6 I=1,N
      IF (IB(I).EQ.0) IB(I)=-1
6     CONTINUE
      RETURN
7     DO 8 I=1,N
      IB(I)=-IB(I)
8     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SORT (SG,R,I2,NVC)
      DIMENSION SG(1), R(1)
C
C     SG IS THE VECTOR OF I2 FUNCTION VALUES TO BE SORTED
C          IN ASCENDING ORDER
C
C     R IS A VECTOR CONTAINING THE VERTEX NUMBERS
C     I2 IS THE NUMBER OF ELEMENTS IN SG
C     NVC IS THE PRESENT NUMBER OF ELEMENTS IN R
C
      NVC1=NVC-I2
      I21=I2-1
      DO 1 I=1,I21
      I1=I+1
      IR=I+NVC1
      DO 1 J=I1,I2
      JR=J+NVC1
      IF (SG(IR).LE.SG(JR)) GO TO 1
      TEMPG=SG(IR)
      TEMPR=R(IR)
      SG(IR)=SG(JR)
      SG(JR)=TEMPG
      R(IR)=R(JR)
      R(JR)=TEMPR
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISTRF (KT,KP,KD,X,Z,I2,I5,NSTEP,DISCR)
      DIMENSION X(1), Z(1), I2(1), I5(1), NSTEP(1), DISCR(KD,1)
      COMMON /TOL4/ ERR(25),ERRO(25)
      COMMON /DISTRA/ KDT,KRD
C
C     THIS SUBROUTINE TRANSFORMS
C        1-THE OPTIMIZATION PARAMETERS BACK TO THE PHYSICAL ONES
C        2-THE DISCRETE VALUES GIVEN BY THE USER IN PERCENTAGE
C          INTO APPROPRIATE VALUES FOR THE DISCRETE OPTIMIZATION
C     AND
C        3-SELECTS AN ERROR MARGIN FOR THE DISCRETE VALUES
C
      DO 2 I=1,KP
      J=I5(I)
      IF (I2(I).EQ.3) GO TO 1
      Z(I)=X(J)
    1 IF (I.LE.KT.AND.I2(I).NE.3) Z(I)=Z(I)**2
    2 CONTINUE
      IF (KRD.EQ.0) GO TO 7
      DO 6 J=1,KRD
      NS=NSTEP(J)
      DO 3 I=1,NS
      DISCR(J,I)=DISCR(J,I)/100.
    3 CONTINUE
      B=DISCR(J,1)
      IF (NS.EQ.1) GO TO 5
      DO 4 K=2,NS
      B=AMIN1(B,DISCR(J,K))
    4 CONTINUE
    5 ERR(J)=0.01*B
    6 CONTINUE
    7 IF (KRD.EQ.KD) RETURN
      KRD1=KRD+1
      DO 10 J=KRD1,KD
      NS=NSTEP(J)
      DO 10 I=1,NS
      B=DISCR(J,1)
      IF (NS.EQ.1) GO TO 9
      DO 8 K=2,NS
      B=AMIN1(B,DISCR(J,K))
    8 CONTINUE
    9 ERR(J)=0.01*B
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COSTFN (X,I5,F,GF,NV,KT,C)
      DIMENSION X(1), I5(1), GF(1), C(1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
C
C     THIS SUBROUTINE COMPUTES THE COST FUNCTION AND ITS DERIVATIVES
C
C     USING THE SAME FUNCTION THE USER CAN SUPPLY HIS OWN COST FUNCTION
C        BY REPLACEMENT.IN CASE OF THE CONTINUOUS PROBLEM THE COST
C        AND ITS DERIVATIVES MUST BE EXPRESSED IN TERMS OF THE
C        TRANSFORMED VARIABLES (Z=X**2)
C
      F=0.
      DO 2 I=1,NV
      IF (I5(I).GT.KT) GO TO 1
      J=I5(I)
      XS=X(I)**2
      IF (KK.EQ.0) F=F+C(J)/XS
      IF (KK.NE.0) F=F+C(J)/X(I)
      IF (KK.EQ.0) GF(I)=-2.*C(J)/(X(I)*XS)
      IF (KK.NE.0) GF(I)=-C(J)/XS
      GO TO 2
1     GF(I)=0.
2     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISCP2 (NR,X,EPS,G,NP,PS,K,NSTEP,DISCR,QSTEP,XB,IX,X1,X
     12,XU,XL,ID,IB,W,H,XE,ICHECK,IVAR,P2,P1,INDX,GPHI,IAA,IBB,GU,PHI,A,
     2T1,T1P,AL,ESTD,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
C
      DIMENSION NSTEP(1), DISCR(K,1), QSTEP(1), IX(1), XB(1), PHI(1), X(
     11), G(1), EPS(1), H(1), W(1), XL(1), ID(1), IB(K,1), IAA(1), IBB(1
     2), XU(1), PS(1), ESTD(1), X1(1), X2(1), XE(NR,NP,1), ICHECK(1), IV
     3AR(1), P2(1), P1(1), INDX(1), GPHI(KP,1), GU(1), A(1), T1(1), T1P(
     41), AL(1)
      DIMENSION Z1(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV1(1), SAM
     1PT(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
      LOGICAL FO
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /DSPT03/ AM,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     10,IFN
      COMMON /DSPT04/ SUMD,INDC
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL1/ KRP1,NVC,NCC
      COMMON /TOL2/ IOP,MOP,IFC,FO
      COMMON /TOL4/ ERF(25),ERRO(25)
      COMMON /TOL5/ NCONS
      COMMON /TOL6/ IALF,NPC
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE SOLVES CONTINUOUS OR DISCRETE PROGRAMMING PROBLEMS
C     THE SOLUTION OF A DISCRETE PROBLEM FOLLOWS THE LOGIC OF DAKINS
C     TREE-SEARCH ALGORITHM
C
C
      INDC=0
      NOR=NR
      N=NOR
      INSOL=0
      IFNT=0
      AL(1)=AO
      INDA=0
      KK=0
      NOD=0
      NORG=NCONS
      PSI=0.
      ESTD(1)=EST1
    1 IF (IOPT1.NE.1.AND.KK.GT.0) GO TO 4
      DO 2 I=1,NOR
      INDX(I)=I
    2 CONTINUE
      IF (IOPT6.EQ.0) GO TO 3
      PA=PS(1)
    3 ICHEK=0
      ALPHA=AL(1)
    4 KKK=0
      IF (IPRINT.GT.0) WRITE (6,104)
      CALL SECOND (T3)
      IF (NCONS.EQ.0) GO TO 20
      IF (IOPT1.EQ.1.AND.NCONS.EQ.1) GO TO 20
      IF (KK.GT.0.AND.IOPT4.EQ.2) IOPT4=1
```

```
      IF (IABS(IOPT4).NE.1) GO TO 5
C
C     FEASIBILITY CHECK, THE VALUE OF P USED IS 2
C
      FO=.F.
      PA=2.
      INDA=1
      CALL DSPTA (N,X,G,H,EPS,1,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,I
     1BB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT
     2,KP,CW)
      KKK=1
      INDA=0
      IF (IFLAGA.EQ.1) GO TO 25
      IF (IOPT4.EQ.-1.AND.IFC.EQ.1) GO TO 95
      IF (IPRINT.GT.0) WRITE (6,104)
C
C     ONE OF THE LEAST PTH OPTIMIZATION ALGORITHMS IS EMPLOYED
C
5     FO=.T.
      IF (IOPT6.EQ.0) GO TO 21
      IF (IOPT6.EQ.2) GO TO 10
      IF (IOPT6.EQ.4) GO TO 6
      IF (IOPT6.EQ.5) GO TO 8
      IF (IOPT6.EQ.3) GO TO 13
      GO TO 21
C
C     NONLINEAR MINIMAX OPTIMIZATION AS A SEQUENCE OF LEAST PTH
C     OPTIMIZATION WITH FINITE VALUES OF P
C
6     IT=1
      PA=PS(1)
7     CALL DSPTA (N,X,G,H,EPS,IT,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,
     1IBB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,K
     2T,KP,CW)
      IF (KO.EQ.0) GO TO 22
      IT=IT+1
      KKK=1
      PSIO=PSI
      PSI=AM+PSIO+1.E-10
      IF (IPRINT.GT.0) WRITE (6,96) PSIO
      IF (IT.EQ.2) GO TO 7
      IF (ABS((PSIO-PSI)/PSIO).GT.ETA) GO TO 7
      GO TO 22
C
C     MODIFIED NON-PARAMETRIC EXTERIOR-POINT METHOD
C
8     IT=1
      PA=PS(1)
      PSI=ESTD(NOD+1)
9     CALL DSPTA (N,X,G,H,EPS,IT,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,
     1IBB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,K
     2T,KP,CW)
      IF (KO.EQ.0) GO TO 22
      IT=IT+1
      KKK=1
      KR=0
      PSIO=PSI
```

```
      PSI=PSIO+SUMD
      IF (IPRINT.GT.0) WRITE (6,97) PSIO
      IF (IT.EQ.2) GO TO 9
      IF (ABS(SUMD/PSIO).GT.ETA) GO TO 9
      GO TO 22
C
C     NONLINEAR MINIMAX OPTIMIZATION AS A SEQUENCE OF LEAST PTH
C     OPTIMIZATION WITH INCREASING VALUES OF P
C
10    DO 12 I=1,NP
      PA=PS(I)
      CALL DSPTA (N,X,G,H,EPS,I,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,I
     1BB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT
     2,KP,CW)
      IF (KO.EQ.0) GO TO 22
      IF (I.LT.2) GO TO 11
      IF (ABS((AMD-AM)/AMD).LE.ETA) GO TO 22
11    AMD=AM
12    CONTINUE
      GO TO 22
C
C     APPLICATION OF AN EXTRAPOLATION TECHNIQUE TO A SEQUENCE OF LEAST
C     PTH OPTIMIZATIONS WITH GEOMETRICALLY INCREASING VALUES OF P
C
13    JORDER=NP-1
      PI=PS(2)/PS(1)
      DO 19 I=1,NP
      PA=PS(I)
      CALL DSPTA (N,X,G,H,EPS,1,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,I
     1BB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT
     2,KP,CW)
      IF (KO.EQ.0) GO TO 22
      CALL DSPTM (NOR,X,XE,I,NP,PI,X1,JORDER)
      IF (I.EQ.1) GO TO 17
      IF (I.EQ.NP) GO TO 15
      DO 14 J=1,NOR
      X2(J)=ABS(X2(J)-X1(J))
      IF (X2(J).GT.EPS(J)*10.) GO TO 17
14    CONTINUE
15    DO 16 J=1,NOR
      X(J)=X1(J)
16    CONTINUE
      GO TO 22
17    DO 18 J=1,NOR
      X2(J)=X1(J)
18    CONTINUE
19    CONTINUE
      GO TO 22
20    PA=PS(1)
C
C     NONLINEAR MINIMAX OPTIMIZATION AS A LEAST PTH OPTIMIZATION WITH A
C     LARGE VALUE OF P
C
21    CALL DSPTA (N,X,G,H,EPS,1,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,I
     1BB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT
     2,KP,CW)
22    CALL DSPTF (X,PHI,U,N,GU,INDX,GPHI,NR,0,Z1,I2,I5,AZ,AX,MU,NV1,SAMP
```

```
     1T,GRAD,PL,PU,W1,KT,KP,CW)
      CALL DSPTD (X,PHI,IAA,IBB,ICHECK,IVAR,P2,P1)
      IFNT=IFNT+1
      IF (IOP.LE.MOP) GO TO 94
      IF (IPRINT.LE.-2) GO TO 26
      IF (IPRINT.EQ.-1.AND.KK.GT.0) GO TO 26
      IF (KK.NE.0) GO TO 24
      DO 23 IZ=1,KP
      W1(IZ)=Z1(IZ)
23    CONTINUE
24    CALL DSPTJ (N,X,F,G,PHI,U,IVAR,W1,I2,I5,KT,KP,NV1,NVC,NPC)
25    IF (IPRINT.LT.0) GO TO 26
      CALL SECOND (T)
      T=T-T3
      WRITE (6,103) T
26    KK=KK+1
      IFNT=IFNT+IFN
C
C     CHECK IF SOLUTION IS FEASIBLE
C
      INDC=0
      IF (IOPT6.EQ.5) ESTD(NOD+2)=U
      IF (IFLAGA.NE.0) GO TO 27
      IF (MAXNOD.EQ.0) GO TO 93
C
C     THE PROBLEM IS SHIFTED TO DEAL WITH THE PHYSICAL PARAMETERS
C
      IF (KK.EQ.1) CALL DISTRF (KT,KP,K,X,Z1,I2,I5,NSTEP,DISCR)
      IF (KK.EQ.1) CALL XZTRAN (Z1,X,I2,I5,NR,KT,K,KP,KR,NPC)
      GO TO 30
27    IF (KK.EQ.1) GO TO 88
C
C     CHECK IF ALTERNATIVE CONSTRAINT AT A PARTICULAR NODE HAS BEEN
C     ADDED
C
28    IF (ICHECK(NOD).EQ.0.OR.ICHECK(NOD).EQ.2) GO TO 29
      ICHECK(NOD)=0
      NCONS=NORG+NOD
      GO TO 1
C
C     CHECK IF ALL NODES HAVE BEEN SEARCHED
C
29    NOD=NOD-1
      IF (NOD.EQ.0) GO TO 87
      GO TO 28
C
C     CHECK IF DISCRETE VALUE SOLUTION IS ATTAINED
C
30    IF (IOPT5.EQ.1) GO TO 31
      GO TO 50
31    DO 41 M=1,K
      IF (ICON.NE.1) GO TO 32
      I=M
      GO TO 33
32    I=K+1-M
33    NS=NSTEP(I)
      IF (X(I).LT.DISCR(I,1)) GO TO 34
```

```
      IF (X(I).GT.DISCR(I,NS)) GO TO 36
      GO TO 38
34    D1=DISCR(I,1)-X(I)
      IF (D1.LE.ERF(I)) GO TO 41
      IF (INSOLN.EQ.0) GO TO 35
      IF (IOPT7.EQ.1) DIFFER=(BSOLN*(1.-SIGN(1.E-6,BSOLN))+ZERO)-U
      IF (IOPT7.NE.1) DIFFER=BSOLN-U
      IF (DIFFER.LT.ZERO) GO TO 28
35    NOD=NOD+1
      ICHECK(NOD)=0
      IVAR(NOD)=I
      P2(NOD)=DISCR(I,1)
      GO TO 44
36    D2=X(I)-DISCR(I,NS)
      IF (D2.LE.ERF(I)) GO TO 41
      IF (INSOLN.EQ.0) GO TO 37
      IF (IOPT7.EQ.1) DIFFER=(BSOLN*(1.-SIGN(1.E-6,BSOLN))+ZERO)-U
      IF (IOPT7.NE.1) DIFFER=BSOLN-U
      IF (DIFFER.LT.ZERO) GO TO 28
37    NOD=NOD+1
      ICHECK(NOD)=2
      IVAR(NOD)=I
      P1(NOD)=DISCR(I,NS)
      GO TO 44
38    NV=NS-1
      DO 39 J=1,NV
      IF (X(I).GE.DISCR(I,J).AND.X(I).LE.DISCR(I,J+1)) GO TO 40
39    CONTINUE
40    D1=X(I)-DISCR(I,J)
      D2=DISCR(I,J+1)-X(I)
      IF (D1.GT.ERF(I).AND.D2.GT.ERF(I)) GO TO 42
41    CONTINUE
      GO TO 77
42    L=I
      LL=J
      IF (INSOLN.EQ.0) GO TO 43
      IF (IOPT7.EQ.1) DIFFER=(BSOLN*(1.-SIGN(1.E-6,BSOLN))+ZERO)-U
      IF (IOPT7.NE.1) DIFFER=BSOLN-U
      IF (DIFFER.LT.ZERO) GO TO 28
43    NOD=NOD+1
      ICHECK(NOD)=1
      IVAR(NOD)=L
      P1(NOD)=DISCR(L,LL)
      P2(NOD)=DISCR(L,LL+1)
44    IF (KK.NE.1.OR.IOPT3.NE.1) GO TO 76
      DO 49 I=1,K
      IF (X(I).LT.DISCR(I,1)) GO TO 46
      IF (X(I).GT.DISCR(I,NS)) GO TO 47
      NS=NSTEP(I)
      DO 45 J=1,NS
      IF (X(I).GE.DISCR(I,J).AND.X(I).LE.DISCR(I,J+1)) GO TO 48
45    CONTINUE
46    XL(I)=DISCR(I,1)
      XU(I)=XL(I)
      GO TO 49
47    XL(I)=DISCR(I,NS)
      XU(I)=XL(I)
```

```
      GO TO 49
48    XL(I)=DISCR(I,J)
      XU(I)=DISCR(I,J+1)
49    CONTINUE
      GO TO 56
50    DO 53 J=1,K
      IF (ICON.NE.1) GO TO 51
      I=J
      GO TO 52
51    I=K+1-J
C
C     X IS INCREASED BY THE TOLERATED ERROR TO GET PROPER DISCRETE
C     VALUES OF X
C
52    ERRO(I)=ERF(I)*QSTEP(I)
      D1=SIGN(ERRO(I),X(I))
      X(I)=X(I)+D1
      IX(I)=IFIX(X(I)/QSTEP(I))
      X(I)=X(I)-D1
      X1(I)=X(I)-FLOAT(IX(I))*GSTEP(I)
      IF (ABS(X1(I)).GT.ERRO(I)) GO TO 54
53    CONTINUE
      GO TO 77
54    L=I
      IF (KK.NE.1.OR.IOPT3.NE.1) GO TO 73
      DO 55 I=1,K
      ERRO(I)=ERF(I)*QSTEP(I)
      D1=SIGN(ERRO(I),X(I))
      X(I)=X(I)+D1
      IX(I)=IFIX(X(I)/QSTEP(I))
      XL(I)=FLOAT(IX(I))*QSTEP(I)
      X(I)=X(I)-D1
55    CONTINUE
C
C     CHECK THE VERTICES ABOUT THE SOLUTION TO THE ORIGINAL
C     CONTINUOUS PROBLEM TO OBTAIN AN INITIAL UPPER BOUND ON THE
C     OBJECTIVE FUNCTION
C
56    NV=2**K
      DO 57 I=1,NV
      ID(I)=I
57    CONTINUE
      DO 59 I=1,NV
      ISUM=1
      DO 58 J=1,K
      M=K+1-J
      MP=2**(M-1)
      IB(M,I)=(ID(I)-ISUM)/MP
      ISUM=ISUM+IB(M,I)*MP
58    CONTINUE
59    CONTINUE
      IF (K.EQ.NOR) GO TO 61
      KP1=K+1
      DO 60 I=KP1,NOR
      X1(I)=X(I)
60    CONTINUE
61    DO 71 I=1,NV
```

```
      IF (IOPT5.EQ.1) GO TO 63
      DO 62 J=1,K
      X1(J)=XL(J) +SIGN(FLOAT(IB(J,I))*QSTEP(J),X(J))
62    CONTINUE
      GO TO 65
63    DO 64 J=1,K
      IF (IB(J,I).EQ.0)  X1(J)=XL(J)
      IF (IB(J,I).EQ.1)  X1(J)=XU(J)
64    CONTINUE
65    CALL DSPTF (X1,PHI,UD,N,GU,INDX,GPHI,NR,0,Z1,I2,I5,AZ,AX,MU,NV1,SA
     1MPT,GRAD,PL,PU,W1,KT,KP,CW)
      IFNT=IFNT+1
      IF (NORG.EQ.0) GO TO 67
      DO 66 J=1,NORG
      IF (PHI(J).LT.ZERO) GO TO 71
66    CONTINUE
67    IF (INSOLN.NE.0) GO TO 70
68    BSOLN=UD
      INSOLN=1
      INSOL=1
      DO 69 J=1,NOR
      XB(J)=X1(J)
69    CONTINUE
      GO TO 71
70    IF (UD.GE.BSOLN) GO TO 71
      GO TO 68
71    CONTINUE
      IF (INSOL.EQ.0) GO TO 72
      KO=1
      IF (IPRINT.GT.-2) CALL DSPTK (BSOLN,XB,PHI,KO,IFNT,IAA,IBB,N,GU,IN
     1DX,GPHI,NR,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT,KP,CW,NVC,
     2NPC)
72    IF (IOPT5.EQ.1) GO TO 76
      GO TO 74
C
C     TERMINATE A NODE IF CORRESPONDING OBJECTIVE FUNCTION VALUE IS
C     WORSE THAN THE BEST DISCRETE VALUE SOLUTION OBTAINED SO FAR
C
73    IF (INSOLN.EQ.0) GO TO 74
      IF (IOPT7.EQ.1) DIFFER=(BSOLN*(1.-SIGN(1.E-6,BSOLN))+ZERO)-U
      IF (IOPT7.NE.1) DIFFER=BSOLN-U
      IF (DIFFER.LT.ZERO) GO TO 28
C
C     IF SOLUTION IS NOT DISCRETE ADD CONSTRAINTS
C
74    NOD=NOD+1
      ICHECK(NOD)=1
      IVAR(NOD)=L
      IF (X(L).LT.0.) GO TO 75
      P1(NOD)=FLOAT(IX(L))*QSTEP(L)
      P2(NOD)=P1(NOD)+QSTEP(L)
      GO TO 76
75    P2(NOD)=FLOAT(IX(L))*QSTEP(L)
      P1(NOD)=P2(NOD)-QSTEP(L)
76    NCONS=NORG+NOD
C
C     CHECK IF MAXIMUM NUMBER OF NODES ALLOWED HAS BEEN EXCEEDED
```

```
C
      IF (KK.GT.MAXNOD) GO TO 89
      GO TO 1
C
C     IF DISCRETE VALUE SOLUTION IS BEST SO FAR RECORD IT
C
77    NNK=INSOLN
      IF (IPRINT.GE.0) WRITE (6,105)
      INSOLN=1
      IF (KK.EQ.1) GO TO 86
      IF (NNK.NE.0) GO TO 85
C
C     NON-ZERO VALUE OF NNK INDICATES THAT AT LEAST A DISCRETE SOLUTION
C     HAS BEEN FOUND
C
78    IF (IOPT5.EQ.1) GO TO 81
      DO 79 I=1,K
      XB(I)=FLOAT(IX(I))*QSTEP(I)
79    CONTINUE
      IF (K.EQ.NOR) GO TO 84
      KP1=K+1
      DO 80 I=KP1,NOR
      XB(I)=X(I)
80    CONTINUE
      GO TO 84
81    DO 82 I=1,K
      XB(I)=X(I)+SIGN(ERF(I),X(I))
      IX(I)=IFIX(XB(I))
      XB(I)=FLOAT(IX(I))
82    CONTINUE
      IF (K.EQ.NOR) GO TO 84
      KP1=K+1
      DO 83 I=KP1,NOR
      XB(I)=X(I)
83    CONTINUE
84    CALL DSPTF (XB,PHI,BSOLN,N,GU,INDX,GPHI,NR,0,Z1,I2,I5,AZ,AX,MU,NV1
     1,SAMPT,GRAD,PL,PJ,W1,KT,KP,CW)
      IFNT=IFNT+1
      INSOL=1
      GO TO 28
85    IF (U.LT.BSOLN) GO TO 78
      GO TO 28
86    KO=0
      IF (IPRINT.GT.-2) CALL DSPTK (U,X,PHI,KO.IFNT,IAA,IBB,N,GU,INDX,GP
     1HI,NR,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT,KP,CW,NVC,NPC)
      RETURN
87    KO=0
      IF (INSOL.EQ.0) GO TO 91
      GO TO 92
88    WRITE (6,98)
      RETURN
89    IF (INSOL.EQ.0) GO TO 90
      WRITE (6,99)
      KO=1
      GO TO 92
90    WRITE (6,100) MAXNOD
      RETURN
```

```
91    WRITE (6,101)
      RETURN
92    IF (IPRINT.GT.-2) CALL DSPTK (BSOLN,XB,PHI,KO,IFNT,IAA,IBB,N,GU,IN
     1DX,GPHI,NR,Z1,I2,I5,AZ,AX,MU,NV1,SAMPT,GRAD,PL,PU,W1,KT,KP,CW,NVC,
     2NPC)
      RETURN
93    WRITE (6,102)
      RETURN
94    CONTINUE
      RETURN
95    WRITE (6,106)
      RETURN
C
C
C
96    FORMAT (1H0,*VALUE OF PSI =*,E14.6)
97    FORMAT (1H0,*ESTIMATE OF MINIMUM ACTUAL FUNCTION VALUE =*,E14.6)
98    FORMAT (1H0,*NO CONTINUOUS SOLUTION, SUGGEST RESTARTING FROM*//1H
     1,*VERY BEGINNING WITH HIGHER VALUE OF ALPHA*)
99    FORMAT (1H0,*MAXIMUM ALLOWABLE NUMBER OF NODES EXCEEDED, BEST DISC
     1RETE SOLUTION IS PRINTED OUT*)
100   FORMAT (1H0,*NO DISCRETE SOLUTION FOUND AFTER*,I5,* NODES*)
101   FORMAT (1H0,*NO DISCRETE SOLUTION*)
102   FORMAT (1H0,*ONLY CONTINUOUS SOLUTION HAS BEEN REQUESTED*)
103   FORMAT (1H0,10X,*EXECUTION TIME IN SECONDS =*,F10.5)
104   FORMAT (1H1)
105   FORMAT (1H0,*THIS IS A DISCRETE SOLUTION*)
106   FORMAT (1H0,//1H ,5X,*THIS FEASIBILITY CHECK CHECKS THE FEASIBILIT
     1Y OF THE*//1H ,5X,*NOMINAL POINT BEFORE FIRST CALL OF VERTST*)
      END
```

```
      SUBROUTINE DSPTA (N,X,G,H,EPS,KR,W,F,ICHECK,IVAR,P2,P1,INDX,GPHI,N
     1R,IAA,IBB,GU,PHI,A,T1,T1P,AL,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU
     2,W1,KT,KP,CW)
C
      DIMENSION W(1), X(1), G(1), H(1), EPS(1), ICHECK(1), IVAR(1), P2(1
     1), P1(1), INDX(1), GPHI(KP,1), IAA(1), IBB(1), GU(1), PHI(1), A(1)
     2, T1(1), T1P(1), AL(1)
      DIMENSION Z1(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMP
     1T(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
      LOGICAL FO
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /DSPT03/ AM,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     1O,IFN
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL1/ KRP1,NVC,NCC
      COMMON /TOL2/ IOP,MOP,IFC,FO
      COMMON /TOL5/ NCONS
      COMMON /TOL6/ IALF,NPC
      COMMON /DEFAULT/ EST,EST1,A0,A1,XMAL,ZERO,ETA,INSOLN,BSOLN
      EXTERNAL DSPTC
C
C     THIS SUBROUTINE COORDINATES THE INPUT, THE OUTPUT AND THE
C     MINIMIZATION
C
      IT=0
      DO 1 I=1,NR
      EPS(I)=EPS(I)/(10.**(KR-1))
    1 CONTINUE
      IF (IOPT1.EQ.1.AND.KK.NE.0) GO TO 2
      GO TO 5
    2 N=NCR-1
      IFV=IVAR(NOD)
      IF (IFV.EQ.NOR) GO TO 4
      TE=EPS(IFV)
      DO 3 I=IFV,NOR
      IP1=I+1
      X(I)=X(IP1)
      EPS(I)=EPS(IP1)
      INDX(I)=IP1
    3 CONTINUE
    4 IF (ICHECK(NOD).EQ.0) X(NOR)=P2(NOD)
      IF (ICHECK(NOD).EQ.1.OR.ICHECK(NOD).EQ.2) X(NOR)=P1(NOD)
    5 KO=1
      ICHEK=0
      IF (KK.GT.1) GO TO 6
      ALPHA=AL(1)
      GO TO 7
    6 IF (ICHECK(NOD).NE.0) ALPHA=AL(NOD)
    7 AL(NOD+1)=ALPHA
      IF (IPRINT.GT.0.AND..NOT.FO) WRITE (6,20)
      IF (IPRINT.GT.0.AND.FO) WRITE (6,18) KR
C
C     SUBROUTINE DSPTB PERFORMS MINIMIZATION BY A VARIABLE METRIC
C     ALGORITHM DUE TO FLETCHER
C
```

```
      IF (IOPT6.EQ.4.AND.KR.EQ.1) GO TO 8
      GO TO 9
8     CALL DSPTC (N,X,F,G,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI,
     1A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      IF (IOPT4.EQ.2) IT=IT+1
      IF (IOPT6.EQ.4.AND.IT.NE.0) PSI=AMIN1(0.,AM+1.E-10)
      IF (IOPT4.NE.2) IT=IT+1
9     ESD=EST
      CALL DSPTB (DSPTC,N,X,ESD,G,H,W,0.,EPS,1,MAX,IPRINT,IEXIT,ICHECK,I
     1VAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI,A,T1,T1P,Z1,I2,I5,AZ,AX,MU,N
     2V,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      ICHEK=1
      KKK=1
C
C     CHECK FEASIBILITY OF CURRENT OPTIMUM SOLUTION
C
      CALL DSPTC (N,X,F,W,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI,
     1A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      IF (IOPT6.EQ.5) GO TO 10
      IF (IFLAGA.EQ.0.OR.IEXIT.EQ.3.OR.INDA.EQ.1) GO TO 10
      KO=0
      ALPHD=ALPHA*AI
      IF (ALPHD.GT.XMAL) GO TO 10
      IF (IOP.LT.MOP) GO TO 17
      ALPHA=ALPHD
      IF (NOD.NE.0) AL(NOD)=ALPHA
      AL(NOD+1)=ALPHA
      KO=1
      GO TO 9
10    IF (IEXIT.EQ.3) KO=0
      IF (INDA.NE.1) GO TO 11
      IF (.NOT.FO.OR.KK.GT.0) KO=2
      INDA=0
      IF (IOPT6.EQ.4) CALL DSPTC (N,X,F,W,ICHECK,IVAR,P2,P1,INDX,GPHI,NR
     1,IAA,IBB,GU,PHI,A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,
     2KT,KP,CW)
11    IF (IOPT1.EQ.1.AND.KK.NE.0.AND.IFV.NE.NOR) GO TO 12
      GO TO 15
12    TS=X(NOR)
      JJ=NOR+1
      NI=NOR-IFV
      DO 13 I=1,NI
      J=JJ-I
      X(J)=X(J-1)
      EPS(J)=EPS(J-1)
13    CONTINUE
      X(IFV)=TS
      EPS(IFV)=TE
      IF (IFV.EQ.NOR) GO TO 15
      DO 14 I=IFV,N
      J=NOR+IFV-I
      G(J)=G(J-1)
14    CONTINUE
15    IF (KO.EQ.2.AND.IPRINT.GT.-1) CALL DSPTJ (N,X,F,G,PHI,U,IVAR,W1,I2
     1,I5,KT,KP,NV,NVC,NPC)
      DO 16 I=1,NR
      EPS(I)=EPS(I)*10.**(KR-1)
```

```
16    CONTINUE
      RETURN
17    ALPHA=ALPHD
      IALF=IALF+1
      IF (IOP.NE.0) IOP=IOP-1
      WRITE (6,19) ALPHA
      RETURN
C
C
C
18    FORMAT (1H0,*OPTIMIZATION*,I3,/,* ---------------*,/,1X,*ITER*,3X,
     1*FUNCT*,8X,*OBJECTIVE*,6X,*VARIABLE*,7X,*GRADIENT*,/)
19    FORMAT (1H0,*ALPHA HAS BEEN INCREASED, ALPHA=*,E14.6,/)
20    FORMAT (1H0,*FEASIBILITY CHECK*,/1H ,17(*-*),/,1X,*ITER*,3X,*FUNCT
     1*,8X,*OBJECTIVE*,6X,*VARIABLE*,7X,*GRADIENT*,/)
      END
```

```
      SUBROUTINE DSPT3 (FUNCT,N,X,F,G,H,W,DFN,EPS,MODE,MAXFN,IPRINT,IEXI
     1T,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI,A,T1,T1P,Z1,I2,I5,
     2AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
C
      DIMENSION X(1), G(1), H(1), W(1), EPS(1), ICHECK(1), IVAR(1), P2(1
     1), P1(1), INDX(1), GPHI(KP,1), IAA(1), IBB(1), GU(1), PHI(1), A(1)
     2, T1(1), T1P(1)
      DIMENSION Z1(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMP
     1T(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
      COMMON /DSPT03/ AM,PSI,PA,XXXXX,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     10,IFN
C
C     UNCONSTRAINED MINIMIZATION METHOD
C
C     R. FLETCHER, FORTRAN SUBROUTINES FOR MINIMIZATION BY QUASI-NEWTON
C     METHODS, ATOMIC ENERGY RESEARCH ESTABLISHMENT, HARWELL, BERKSHIRE,
C     ENGLAND, REPORT AERE- R7125, 1972
C
      IF (KKK.NE.0) GO TO 1
      ITN=0
      IFN=1
    1 CONTINUE
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=0
      IF (MODE.EQ.3) GO TO 7
      IF (MODE.EQ.2) GO TO 4
      IJ=NN+1
      DO 3 I=1,N
      DO 2 J=1,I
      IJ=IJ-1
      H(IJ)=0.
    2 CONTINUE
      H(IJ)=1.
    3 CONTINUE
      GO TO 7
    4 CONTINUE
      IJ=1
      DO 6 I=2,N
      Z=H(IJ)
      IF (Z.LE.0.) RETURN
      IJ=IJ+1
      I1=IJ
      DO 6 J=I,N
      ZZ=H(IJ)
      H(IJ)=H(IJ)/Z
      JK=IJ
      IK=I1
      DO 5 K=I,J
      JK=JK+NP-K
      H(JK)=H(JK)-H(IK)*ZZ
      IK=IK+1
```

```
5     CONTINUE
6     IJ=IJ+1
      IF (H(IJ).LE.0.) RETURN
7     CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 8 I=2,N
      IF (H(IJ).GE.DMIN) GO TO 8
      DMIN=H(IJ)
8     IJ=IJ+NP-I
      IF (DMIN.LE.0.) RETURN
      Z=F
      CALL FUNCT (N,X,F,G,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI,
     1A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      IF (INDB.EQ.1) GO TO 37
      DF=DFN
      IF (DFN.EQ.0.) DF=F-Z
      IF (DFN.LT.0.) DF=ABS(DF*F)
      IF (DF.LE.0.) DF=1.
9     CONTINUE
      IF (IPRINT.LE.0) GO TO 10
      IF (MOD(ITN,IPRINT).NE.0) GO TO 10
      PRINT 38, ITN,IFN,F,((X(I),G(I)),I=1,N)
10    CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 12 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 11 J=1,I1
      Z=Z-H(IJ)*W(J)
      IJ=IJ+N-J
11    CONTINUE
      W(I)=Z
12    CONTINUE
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 14 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 13 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
      IJ=IJ-1
13    CONTINUE
      W(IS+N-I)=W(N-I)/H(IJ)-Z
14    CONTINUE
      GS=0.
      DO 15 I=1,N
      GS=GS+W(IS+I)*G(I)
15    CONTINUE
      IEXIT=2
      IF (GS.GE.0.) GO TO 37
      GS0=GS
      ALPHA=-2.*DF/GS
      IF (ALPHA.GT.1.) ALPHA=1.
      DF=F
```

```
      TOT=0.
16    CONTINUE
      IEXIT=3
      IF (IFN.EQ.MAXFN) GO TO 37
      ICON=0
      IEXIT=1
      DO 17 I=1,N
      Z=ALPHA*W(IS+I)
      IF (ABS(Z).GE.EPS(I)) ICON=1
      X(I)=X(I)+Z
17    CONTINUE
      CALL FUNCT (N,X,FY,W,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,GU,PHI
     1,A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
      IF (INDB.EQ.1) GO TO 37
      IFN=IFN+1
      GYS=0.
      DO 18 I=1,N
      GYS=GYS+W(I)*W(IS+I)
18    CONTINUE
      IF (FY.GE.F) GO TO 19
      IF (ABS(GYS/GS0).LE..9) GO TO 21
      IF (GYS.GT.0.) GO TO 19
      TOT=TOT+ALPHA
      Z=10.
      IF (GS.LT.GYS) Z=GYS/(GS-GYS)
      IF (Z.GT.10.) Z=10.
      ALPHA=ALPHA*Z
      F=FY
      GS=GYS
      GO TO 16
19    CONTINUE
      DO 20 I=1,N
      X(I)=X(I)-ALPHA*W(IS+I)
20    CONTINUE
      IF (ICON.EQ.0) GO TO 37
      Z=3.*(F-FY)/ALPHA+GYS+GS
      ZZ=SQRT(Z**2-GS*GYS)
      Z=1.-(GYS+ZZ-Z)/(2.*ZZ+GYS-GS)
      ALPHA=ALPHA*Z
      GO TO 16
21    CONTINUE
      ALPHA=TOT+ALPHA
      F=FY
      IF (ICON.EQ.0) GO TO 35
      DF=DF-F
      DGS=GYS-GS0
      LINK=1
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 23
      DO 22 I=1,N
      W(IU+I)=W(I)-G(I)
22    CONTINUE
      SIG=1./(ALPHA*DGS)
      GO TO 30
23    CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
      DO 24 I=1,N
```

```
      W(IU+I)=Z*G(I)+W(I)
   24 CONTINUE
      SIG=1./(ZZ*DGS**2)
      GO TO 30
   25 CONTINUE
      LINK=2
      DO 26 I=1,N
      W(IU+I)=G(I)
   26 CONTINUE
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 27
      SIG=1./GS0
      GO TO 30
   27 CONTINUE
      SIG=-ZZ
      GO TO 30
   28 CONTINUE
      DO 29 I=1,N
      G(I)=W(I)
   29 CONTINUE
      GO TO 9
   30 CONTINUE
      W(IV+1)=W(IU+1)
      DO 32 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 31 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
   31 CONTINUE
      W(IV+I)=Z
   32 CONTINUE
      IJ=1
      DO 33 I=1,N
      Z=H(IJ)+SIG*W(IV+I)**2
      IF (Z.LE.0.) Z=DMIN
      IF (Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IB+I)=W(IV+I)*SIG/Z
      SIG=SIG-W(IB+I)**2*Z
      IJ=IJ+NP-I
   33 CONTINUE
      IJ=1
      DO 34 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 34 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
   34 IJ=IJ+1
      GO TO (25,28), LINK
   35 CONTINUE
      DO 36 I=1,N
      G(I)=W(I)
   36 CONTINUE
   37 CONTINUE
      IF (IPRINT.LE.0) RETURN
```

```
      PRINT 38, ITN,IFN,F,((X(I),G(I)),I=1,N)
      IF (INDB.EQ.1) RETURN
      PRINT 39, IEXIT
      IF (IEXIT.EQ.1) PRINT 40
      IF (IEXIT.EQ.2) PRINT 41
      IF (IEXIT.EQ.3) PRINT 42
C
      RETURN
C
C
C
38    FORMAT (1H ,I4,3X,I4,6X,E14.6,1X,60(E14.6,1X,E14.6,/,33X))
39    FORMAT (1H0,*IEXIT =*,I5)
40    FORMAT (1H0,*NORMAL EXIT*)
41    FORMAT (1H0,*EPS IS PROBABLY SET TOO SMALL*)
42    FORMAT (1H0,*PERMISSIBLE NUMBER OF FUNCTION EVALUATIONS EXCEEDED*)
      END
```

```
      SUBROUTINE DSPTC (N,Y,F,G,ICHECK,IVAR,P2,P1,INDX,GPHI,NR,IAA,IBB,G
     1U,PHI,A,T1,T1P,Z1,I2,I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW)
C
      DIMENSION Y(1), G(1), IAA(1), IBB(1), GU(1), PHI(1), GPHI(KP,1), I
     1CHECK(1), IVAR(1), P2(1), P1(1), A(1), T1(1), T1P(1), INDX(1)
      DIMENSION Z1(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMP
     1T(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /DSPT03/ AM,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     1O,IFN
      COMMON /DSPT04/ SUMD,INDC
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE COORDINATES SUBROUTINES DSPTD,DSPTE,DSPTF,DSPTG
C     AND DSPTH TO FORMULATE THE ARTIFICIAL UNCONSTRAINED OBJECTIVE
C     FUNCTION AND THE NECESSARY GRADIENTS
C
      IDD=0
      IFV=IVAR(NOD)
      IF (IOPT1.EQ.1.AND.KK.GT.0.AND.IFV.NE.NOR) GO TO 1
      GO TO 3
1     IDD=1
      TS=Y(NOR)
      NI=NOR-IFV
      JJ=NOR+1
      DO 2 I=1,NI
      J=JJ-I
      Y(J)=Y(J-1)
2     CONTINUE
      Y(IFV)=TS
3     CALL DSPTF (Y,PHI,U,N,GU,INDX,GPHI,NR,1,Z1,I2,I5,AZ,AX,MU,NV,SAMPT
     1,GRAD,PL,PU,W1,KT,KP,CW)
      CALL DSPTD (Y,PHI,IAA,IBB,ICHECK,IVAR,P2,P1)
      IF (KK.EQ.0.OR.INDC.EQ.1) GO TO 4
      IF (IOPT1.EQ.1.AND.NOD.LE.1) GO TO 4
      CALL DSPTE (IAA,IBB,GPHI,NR,IVAR,KP)
4     IF (IOPT1.NE.1.OR.KK.EQ.0.OR.IFV.EQ.NOR) GO TO 8
      IF (INDC.EQ.1) NCM1=NORG
      IF (INDC.NE.1) NCM1=NCONS-1
      DO 6 J=1,NCM1
      DO 5 I=IFV,N
      GPHI(I,J)=GPHI(I+1,J)
5     CONTINUE
6     CONTINUE
      DO 7 I=IFV,N
      GU(I)=GU(I+1)
7     CONTINUE
8     CALL DSPTH (U,GU,PHI,N,F,G,A,T1,T1P,NR,GPHI,KP)
      IF (IDD.EQ.0) GO TO 10
      TTS=Y(IFV)
      DO 9 I=IFV,N
      Y(I)=Y(I+1)
9     CONTINUE
      Y(NOR)=TTS
10    IF (KK.NE.0) INDC=1
      RETURN
      END
```

```
      SUBROUTINE DSPTD (X,PHI,IAA,IBB,ICHECK,IVAR,P2,P1)
C
      DIMENSION X(1), PHI(1), IAA(1), IBB(1), ICHECK(1), IVAR(1), P2(1),
     1 P1(1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE RETURNS ADDITIONAL PARAMETER CONSTRAINTS FOR
C     DISCRETE VALUE OPTIMIZATION
C
      IF (NOD.EQ.0) GO TO 4
      IF (NOD.EQ.1.AND.IOPT1.EQ.1) GO TO 4
      MN=NOD
      IF (IOPT1.EQ.1) MN=NOD-1
      DO 3 I=1,MN
      L=IVAR(I)
      II=I+NORG
      IF (L.EQ.IVAR(NOD).AND.IOPT1.EQ.1) GO TO 2
      IF (ICHECK(I).EQ.0) GO TO 1
      PHI(II)=P1(I)-X(L)
      IAA(II)=-1
      IBB(II)=L
      GO TO 3
    1 PHI(II)=(X(L)-P2(I))
      IAA(II)=1
      IBB(II)=L
      GO TO 3
    2 PHI(II)=1.E+10
      IAA(II)=0
    3 CONTINUE
    4 RETURN
      END
```

```
      SUBROUTINE DSPTE (IAA,IBB,GPHI,NR,IVAR,KP)
C
      DIMENSION IAA(1), IBB(1), GPHI(KP,1), IVAR(1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE RETURNS THE GRADIENTS OF THE ADDITIONAL PARAMETER
C     CONSTRAINTS FOR DISCRETE VALUE OPTIMIZATION
C
      NORGP=NORG+1
      MN=NCONS
      IF (IOPT1.EQ.1) MN=NCONS-1
      DO 3 J=1,NOR
      IF (IOPT1.EQ.1.AND.J.EQ.IVAR(NOD)) GO TO 3
      DO 2 I=NORGP,MN
      IF (IBB(I).NE.J) GO TO 1
      GPHI(J,I)=IAA(I)
      GO TO 2
1     GPHI(J,I)=0.
2     CONTINUE
3     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DSPTF (X,G,F,N,GF,INDX,GG,NR,IG,Z,I2,I5,AZ,AX,MU,NV,SAM
     1PT,GRAD,PL,PU,W1,KT,KP,CW)
      DIMENSION X(1), G(1), GG(KP,1), GF(1), INDX(1)
      DIMENSION Z(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMPT
     1(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
C
C     THIS SUBROUTINE COORDINATE THE OBJECTIVE FUNCTION AND ITS
C        DERIVATIVES,THE CONSTRAINT FUNCTIONS AND THEIR DERIVATIVES
C        FOR USE IN DISOP2
C
      CALL COSTFN (X,I5,F,GF,NR,KT,CW)
      CALL CONSTR (Z,X,I2,I5,AZ,AX,MU,G,GG,NV,SAMPT,GRAD,PL,PU,W1,NR,KT,
     1KP)
      RETURN
      END
```

```
      SUBROUTINE DSPTH (U,GU,PHI,N,F,G,A,T1,T1P,NR,GPHI,KP)
C
      DIMENSION GU(1), PHI(1), G(1), A(1), T1(1), T1P(1), GPHI(KP,1)
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /DSPT03/ AMD,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,
     1KO,IFN
      COMMCN /DSPT04/ SUMD,INDC
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE TRANSFORMS THE CONSTRAINED PROBLEM INTO AN
C     UNCONSTRAINED OBJECTIVE USING THE BANDLER-CHARALAMBOUS TECHNIQUE
C     OR A MODIFIED NON-PARAMETRIC EXTERIOR-POINT METHOD
C
      EPSPHI=-ZERO
      P=PA
      AE=0.
      UR=U
      NC=NCONS
      IFLAGA=0
      INDB=0
      IF (NCONS.EQ.0.OR.ALPHA.EQ.0.) GO TO 23
      NCM=NCONS
      IF (IOPT1.EQ.1.AND.KK.GT.0) NCM=NCONS-1
      IF (NCM.EQ.0) GO TO 23
      NT=NCM+1
      IF (INDA.EQ.1.OR.IOPT6.EQ.5) GO TO 2
      V=(U-PSI)/ALPHA
      DO 1 I=1,NCM
      A(I)=V-PHI(I)
    1 CONTINUE
      A(NT)=V
      GO TO 6
    2 DO 3 I=1,NCM
      A(I)=-PHI(I)
    3 CONTINUE
      IF (INSOLN.EQ.1.AND.INDA.EQ.1) GO TO 4
      A(NT)=-1.E+20
      GO TO 5
    4 IF (IOPT7.EQ.1) A(NT)=U-(BSOLN*(1.-SIGN(ETA,BSOLN))+ZERO)
      IF (IOPT7.NE.1) A(NT)=U-BSOLN
    5 IF (IOPT6.EQ.5.AND.INDA.NE.1) A(NT)=U-PSI
    6 AM=A(1)
      DO 7 I=2,NT
      AM=AMAX1(AM,A(I))
    7 CONTINUE
      IF (INDA.NE.1.AND.IOPT6.NE.5) AMD=AM*ALPHA
      IF (INDA.EQ.1.AND.AM.LE.EPSPHI) INDB=1
      IF (AM.LE.0.) P=-PA
      SUM1=0.
      DO 11 I=1,NT
      IF (AM) 10,8,9
    8 AE=1.E-10
      GO TO 10
```

```
9     IF (A(I).LE.0.) GO TO 11
10    T1(I)=(A(I)-AE)/(AM-AE)
      T1P(I)=T1(I)**P
      SUM1=SUM1+T1P(I)
11    CONTINUE
      SUM3=SUM1**(1./P)
      F=(AM-AE)*SUM3
      IF (IOPT6.NE.5.AND.INDA.NE.1) F=F*ALPHA
      SUMD=F
      DO 22 I=1,N
      SUM2=0.
      IF (INDA.EQ.1.OR.IOPT6.EQ.5) GO TO 16
      DO 15 J=1,NT
      IF (AM) 13,13,12
12    IF (A(J).LE.0.) GO TO 15
13    IF (J.EQ.NT) GO TO 14
      SUM2=SUM2+T1P(J)/T1(J)*(GU(I)/ALPHA-GPHI(I,J))
      GO TO 15
14    SUM2=SUM2+T1P(J)/T1(J)*GU(I)/ALPHA
15    CONTINUE
      SUM2=SUM2*ALPHA
      GO TO 21
16    DO 20 J=1,NT
      IF (AM) 18,18,17
17    IF (A(J).LE.0.) GO TO 20
18    IF (J.EQ.NT) GO TO 19
      SUM2=SUM2-T1P(J)/T1(J)*GPHI(I,J)
      GO TO 20
19    IF (IOPT6.NE.5.AND.INSOLN.NE.INDA) GO TO 20
      SUM2=SUM2+T1P(J)/T1(J)*GU(I)
20    CONTINUE
21    G(I)=SUM3/SUM1*SUM2
22    CONTINUE
      GO TO 25
23    F=U
      DO 24 I=1,N
      G(I)=GU(I)
24    CONTINUE
25    IF (ICHEK.EQ.0.OR.NCONS.EQ.0.OR.NCM.EQ.0) RETURN
      DO 26 I=1,NCM
      IF (PHI(I).LT.ZERO) IFLAGA=1
26    CONTINUE
      IF (INDA.NE.1) RETURN
      IF (A(NT).LE.EPSPHI) RETURN
      IFLAGA=1
      RETURN
      END
```

```
      SUBROUTINE DSPTI (KP,KT,N,K,Z,I2,W,EPS,PS,NP,QSTEP,NSTEP,DISCR,SAM
     1PT,GG)
      DIMENSION EPS(1), PS(1), QSTEP(1), NSTEP(1), DISCR(K,1), Z(1), W(1
     1), SAMPT(3,1), I2(1), GG(KP,1)
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE PRINTS THE INPUT DATA
C
      WRITE (6,10)
      IF (IOPT5.NE.1) GO TO 3
      J=1
      DO 1 I=1,KP
      IF (I2(I).NE.1) GO TO 1
      W(J)=FLOAT(I)
      J=J+1
1     CONTINUE
      WRITE (6,11)
      DO 2 I=1,K
      II=IFIX(W(I))
      NS=NSTEP(I)
      WRITE (6,12) II,(DISCR(I,J),J=1,NS)
2     CONTINUE
3     CONTINUE
      IF (I2(1).EQ.1) WRITE (6,15) GG(1,1)
      IF (I2(1).EQ.2) WRITE (6,16) GG(1,1)
      IF (I2(1).EQ.3) WRITE (6,17) GG(1,1)
      DO 4 I=2,KP
      II=I2(I)
      IF (II.EQ.1.AND.I.LE.KT) WRITE (6,18) I,GG(I,1)
      IF (II.EQ.1.AND.I.GT.KT) WRITE (6,21) I,GG(I,1)
      IF (II.EQ.2.AND.I.LE.KT) WRITE (6,19) I,GG(I,1)
      IF (II.EQ.2.AND.I.GT.KT) WRITE (6,22) I,GG(I,1)
      IF (II.EQ.3.AND.I.LE.KT) WRITE (6,20) I,GG(I,1)
      IF (II.EQ.3.AND.I.GT.KT) WRITE (6,23) I,GG(I,1)
4     CONTINUE
      IF (IOPT6.NE.0) WRITE (6,30) ZERO
      IF (MAXNOD.EQ.0) GO TO 5
      IF (IOPT5.EQ.1) GO TO 5
      WRITE (6,51) QSTEP(1)
      IF (K.LT.2) GO TO 5
      WRITE (6,52) (I,QSTEP(I),I=2,K)
5     WRITE (6,24) EPS(1)
      WRITE (6,25) (I,EPS(I),I=2,N)
      WRITE (6,26) EST
      IF (IOPT6.EQ.0) RETURN
      IF (IOPT6.EQ.5) WRITE (6,50) EST1
      IF (INSOLN.EQ.1) WRITE (6,47) BSOLN
      IF (IOPT6.EQ.5) GO TO 6
      WRITE (6,27) AO
      WRITE (6,29) XMAL
      WRITE (6,31) AI
6     IF (IOPT6.EQ.4.OR.IOPT6.EQ.2.OR.IOPT6.EQ.5) WRITE (6,38) ETA
      WRITE (6,42) NP
      WRITE (6,43) PS(1)
```

```
      IF (NP.LT.2) GO TO 7
      WRITE (6,44) (I,PS(I),I=2,NP)
      WRITE (6,13) MAX
      IF (MAXNOD.NE.0) WRITE (6,28) MAXNOD
      WRITE (6,14) IPRINT
7     WRITE (6,32)
      IF (IOPT6.EQ.4) WRITE (6,39)
      IF (IOPT6.EQ.1) WRITE (6,41)
      IF (IOPT6.EQ.2) WRITE (6,40)
      IF (IOPT6.EQ.5) WRITE (6,46)
      IF (IOPT6.EQ.3) WRITE (6,48)
      IF (MAXNOD.EQ.0) GO TO 8
      IF (IOPT1.EQ.1) WRITE (6,33)
      IF (IOPT3.EQ.1) WRITE (6,34)
      IF (IOPT4.EQ.1) WRITE (6,35)
      IF (IOPT4.EQ.2) WRITE (6,45)
      IF (IOPT7.EQ.1) WRITE (6,49)
      IF (ICON.EQ.1) WRITE (6,36)
      IF (ICON.NE.1) WRITE (6,37)
8     WRITE (6,53)
      DO 9 IF=1,NSP
      WRITE (6,54) IF,SAMPT(1,IF),SAMPT(2,IF),SAMPT(3,IF)
9     CONTINUE
      RETURN
C
C
C
10    FORMAT (1H1,*INPUT DATA*,/,1X,10(*-*),//)
11    FORMAT (* DISCRETE VALUES FOR THE VARIABLES*)
12    FORMAT (1H0,*Z(*,I2,*)*,5E16.8)
13    FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE FUNCTION EVALUATIONS*,17(
     1*.*),*MAX=*,I5)
14    FORMAT (1H0,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERAT
     1IONS*,5(*.*),*IPRINT=*,I5)
15    FORMAT (1H0,*USER SUPPLIED COMPONENTS*,8(*.*),*TOLERANCE,DISCRETE*
     1,13(*.*),*Z( 1)=*,E16.8)
16    FORMAT (1H0,*USER SUPPLIED COMPONENTS*,8(*.*),*TOLERANCE,CONTINUOU
     1S*,11(*.*),*Z( 1)=*,E16.8)
17    FORMAT (1H0,*USER SUPPLIED COMPONENTS*,8(*.*),*TOLERANCE,FIXED*,16
     1(*.*),*Z( 1)=*,E16.8)
18    FORMAT (33X,*TOLERANCE,DISCRETE*,13(*.*),*Z(*,I2,*)=*,E16.8)
19    FORMAT (33X,*TOLERANCE,CONTINUOUS*,11(*.*),*Z(*,I2,*)=*,E16.8)
20    FORMAT (33X,*TOLERANCE,FIXED*,16(*.*),*Z(*,I2,*)=*,E16.8)
21    FORMAT (33X,*NOMINAL,DISCRETE*,15(*.*),*Z(*,I2,*)=*,E16.8)
22    FORMAT (33X,*NOMINAL,CONTINUOUS*,13(*.*),*Z(*,I2,*)=*,E16.8)
23    FORMAT (33X,*NOMINAL,FIXED*,18(*.*),*Z(*,I2,*)=*,E16.8)
24    FORMAT (1H0,*TEST QUANTITIES TO BE USED IN FLETCHER METHOD*,16(*.*
     1),*EPS( 1)=*,E16.8)
25    FORMAT (62X,*EPS(*,I2,*)=*,E16.8)
26    FORMAT (1H0,*ESTIMATE OF LOWER BOUND ON ARTIFICIAL OBJECTIVE FUNCT
     1ION*,9(*.*),*EST=*,E16.8)
27    FORMAT (1H0,*INITIAL VALUE OF THE PARAMETER ALPHA*,30(*.*),*AO=*,E
     116.8)
28    FORMAT (1H0,*MAXIMUM NUMBER OF NODES TO BE SEARCHED*,24(*.*),*MAXN
     1OD=*,I5)
29    FORMAT (1H0,*MAXIMUM ALLOWABLE VALUE OF THE PARAMETER ALPHA*,18(*.
     1*),*XMAL=*,E16.8)
```

```
30    FORMAT (1H0,*ERROR TOLERANCE IN CONSTRAINTS*,34(*.*),*ZERO=*,E16.8
     1)
31    FORMAT (1H0,*MULTIPLYING FACTOR IN ALPHA VALUE*,33(*.*),*AI=*,E16.
     18)
32    FORMAT (1H0,*FOLLOWING OPTIONS USED*,/,1X,22(*-*),/)
33    FORMAT (1H0,*(N-1) VARIABLE OPTIMIZATION PERFORMED IN DISCRETE PRO
     1BLEM*)
34    FORMAT (1H0,*VERTICES CHECKED ABOUT CONTINUOUS SOLUTION TO OBTAIN
     1AN INITIAL UPPER BOUND IN*/1H ,*DISCRETE PROBLEM*)
35    FORMAT (1H0,*FEASIBILITY CHECKED IN FINAL OPTIMIZATION*)
36    FORMAT (1H0,*PARTITIONING STARTS ON FIRST DISCRETE VARIABLE*)
37    FORMAT (1H0,*PARTITIONING STARTS ON LAST DISCRETE VARIABLE*)
38    FORMAT (1H0,*TEST QUANTITY TO BE USED IN NLP ALGORITHM 2/4/5*,18(*
     1.*),*ETA=*,E16.8)
39    FORMAT (1H0,*NLP ALGORITHM 4 EMPLOYED-SEQUENCE OF LEAST PTH OPTIMI
     1ZATION WITH FINITE VALUES OF P*)
40    FORMAT (1H0,*NLP ALGORITHM 2 EMPLOYED-SEQUENCE OF LEAST PTH OPTIMI
     1ZATION WITH INCREASING VALUES OF P*)
41    FORMAT (1H0,*NLP ALGORITHM 1 EMPLOYED-ONE LEAST PTH OPTIMIZATION W
     1ITH A LARGE VALUE OF P*)
42    FORMAT (1H0,*NUMBER OF P VALUES*,48(*.*),*NP=*,I5)
43    FORMAT (1H0,*VALUE(S) OF P USED IN NLP ALGORITHM*,27(*.*),*PS( 1)=
     1*,E16.8)
44    FORMAT (63X,*PS(*,I2,*)=*,E16.8)
45    FORMAT (1H0,*FEASIBILITY CHECKED IN FINAL OPTIMIZATION FOR DISCRET
     1E PROBLEM ONLY*)
46    FORMAT (1H0,*NLP ALGORITHM 5 EMPLOYED- A MODIFIED NON-PARAMETRIC E
     1XTERIOR-POINT METHOD*)
47    FORMAT (1H0,*UPPER BOUND ON ARTIFICIAL OBJECTIVE FUNCTION*,21(*.*)
     1,*IDS=*,E16.8)
48    FORMAT (1H0,*NLP ALGORITHM 3 EMPLOYED- APPLICATION OF EXTRAPOLATIO
     1N TECHNIQUE*)
49    FORMAT (1H0,*ONLY ONE OPTIMUM DISCRETE SOLUTION REQUIRED*)
50    FORMAT (1H0,*ESTIMATE OF LOWER BOUND ON ACTUAL OBJECTIVE FUNCTION*
     1,12(*.*),*EST1=*,E16.8)
51    FORMAT (1H0,*QUANTIZATION STEP SIZES FOR THE DISCRETE VARIABLES*,9
     1(*.*),*QSTEP( 1)=*,E16.8)
52    FORMAT (60X,*QSTEP(*,I2,*)=*,E16.8)
53    FORMAT (//1H0,*DATA GIVEN FOR SPECIFIC PROBLEM*,/,1X,31(*-*),/1H0,
     19X,*IF*,7X,*SAMPT(1,IF)*,7X,*SAMPT(2,IF)*,7X,*SAMPT(3,IF)*,/)
54    FORMAT (6X,I3,3E18.6)
      END
```

```
      SUBROUTINE DSPTJ (N,X,F,G,PHI,U,IVAR,Z,I2,I5,KT,KP,NV,NVC,NPC)
C
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /DSPT03/ AM,PSI,PA,ALPHA,IFLAGA,ICHEK,KKK,INDA,INDB,UR,NC,K
     10,NUMF
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
      DIMENSION X(1), G(1), PHI(1), IVAR(1)
      DIMENSION Z(1), I2(1), I5(1), NV(1)
C
C     THIS SUBROUTINE OUTPUTS THE SOLUTION AT EACH NODE
C
      IF (IOPT1.EQ.1.AND.KK.GT.0) GO TO 1
      NCM=NC
      NM=N
      GO TO 2
1     NCM=NC-1
      NM=N+1
2     NVIOL=0
      IF (KO.EQ.0) WRITE (6,28)
      IF (KO.EQ.1.AND.KK.EQ.0) WRITE (6,36)
      IF (KO.EQ.1.AND.KK.GT.0) WRITE (6,27)
      IF (KO.EQ.2) WRITE (6,30)
      IF (MAXNOD.NE.0) WRITE (6,34) KK
      IF (KO.EQ.2) GO TO 7
      IF (IOPT6.NE.3) WRITE (6,29) F
      WRITE (6,18) U
      IF (IOPT6.NE.3) GO TO 4
      DO 3 I=1,NM
      WRITE (6,35) I,X(I)
3     CONTINUE
      GO TO 7
4     DO 6 I=1,NM
      IF (IOPT1.EQ.1.AND.I.EQ.IVAR(NOD)) GO TO 5
      WRITE (6,31) I,X(I),I,G(I)
      GO TO 6
5     WRITE (6,32) I,X(I)
6     CONTINUE
7     IF (NCM.EQ.0) GO TO 15
      DO 8 I=1,NCM
      IF (PHI(I).LT.ZERO) NVIOL=NVIOL+1
8     CONTINUE
      WRITE (6,19)
      IND=0
      IND1=1
      DO 10 I=1,NVC
      IND=IND+1
      IF (IND.LE.NV(IND1)) GO TO 9
      IND=1
      IND1=IND1+1
9     WRITE (6,20) I,PHI(I),IND1
10    CONTINUE
      IF (NPC.EQ.0) GO TO 12
      II=NVC
```

```
      DO 11 I=1,NPC
      II=II+1
      WRITE (6,21) II,PHI(II),I
      II=II+1
      WRITE (6,22) II,PHI(II),I
11    CONTINUE
12    CONTINUE
      IF (NEC.EQ.0) GO TO 14
      DO 13 I=1,NEC
      II=II+1
      WRITE (6,23) II,PHI(II),I
13    CONTINUE
14    CONTINUE
      IF (MAXNOD.NE.0) WRITE (6,24) NCM
      WRITE (6,33) NVIOL
15    WRITE (6,25) NUMF
      IF (KO.EQ.2) RETURN
      IF (IOPT6.NE.5) WRITE (6,26) ALPHA
      IF (KK.GT.0) RETURN
      DO 17 I=1,KP
      J=I5(I)
      IF (I2(I).EQ.3) GO TO 16
      Z(I)=X(J)
16    IF (I.LE.KT.AND.I2(I).NE.3) Z(I)=Z(I)**2
17    CONTINUE
      WRITE (6,27)
      WRITE (6,37) (I,Z(I),I=1,KP)
      RETURN
C
C
C
18    FORMAT (1H0,8X,*ACTUAL OBJECTIVE FUNCTION F =*,E16.8,/)
19    FORMAT (1H0,7X,*INEQUALITY CONSTRAINTS*,5X,*OCCURRING AT*,/)
20    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*SAMPLE POINT*,I2)
21    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*LOWER BOUND*,I3)
22    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*UPPER BOUND*,I3)
23    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*EXTRA CONST*,I3)
24    FORMAT (1H0,9X,*NUMBER OF CONSTRAINTS USED =*,I5)
25    FORMAT (1H0,5X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
26    FORMAT (1H0,1X,*FINAL VALUE OF THE PARAMETER ALPHA =*,E16.8)
27    FORMAT (1H1,11X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,12X,*-------
     1-------------------------*)
28    FORMAT (1H1,15X,*RESULTS AT LAST ITERATION*/,16X,*----------------
     1---------*)
29    FORMAT (1H0,*ARTIFICIAL UNCONSTRAINED FUNCTION U =*,E16.8)
30    FORMAT (1H1,10X,*RESULTS OF THE FEASIBILITY CHECK*,/,11X,32(*-*))
31    FORMAT (8X,*X(*,I2,*)=*,E16.8,1X,*GU(*,I2,*)=*,E16.8)
32    FORMAT (8X,*X(*,I2,*)=*,E16.8)
33    FORMAT (1H0,5X,*NUMBER OF VIOLATED CONSTRAINTS =*,I5)
34    FORMAT (1H0,24X,*NODE NUMBER =*,I5)
35    FORMAT (8X,*X(*,I2,*)=*,E16.8)
36    FORMAT (1H0,11X,*FOLLOWING IS RESULT OF OPTIMIZATION*,/,12X,35(*-*
     1),/)
37    FORMAT (8X,*Z(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE DSPTK (U,X,PHI,KO,IFNT,IAA,IBB,N,GU,INDX,GPHI,NR,Z1,I2,
     1I5,AZ,AX,MU,NV,SAMPT,GRAD,PL,PU,W1,KT,KP,CW,NVC,NPC)
      DIMENSION X(1), PHI(1), IAA(1), IBB(1), GU(1), INDX(1), GPHI(KP,1)
      DIMENSION Z1(1), I2(1), I5(1), AZ(1), AX(1), MU(KT,1), NV(1), SAMP
     1T(3,1), GRAD(1), PL(1), PU(1), W1(1), CW(1)
C
      COMMON /DSPT02/ NOD,KK,NORG,NOR
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOP
     1T4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXV
     2N,NVSUM,NEC,ND1,ND6
      COMMON /TOL5/ NCONS
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
C
C     THIS SUBROUTINE OUTPUTS THE FINAL SOLUTION IN A STANDARD FORM
C
      WRITE (6,20)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,10)
      WRITE (6,11) U
      GO TO 2
1     WRITE (6,12)
      WRITE (6,13) U
2     WRITE (6,14) (I,X(I),I=1,NOR)
      IF (NORG.EQ.0) GO TO 9
      CALL DSPTF (X,PHI,U,N,GU,INDX,GPHI,NR,0,Z1,I2,I5,AZ,AX,MU,NV,SAMPT
     1,GRAD,PL,PU,W1,KT,KP,CW)
      IFNT=IFNT+1
      WRITE (6,15)
      IND=0
      IND1=1
      DO 4 I=1,NVC
      IND=IND+1
      IF (IND.LE.NV(IND1)) GO TO 3
      IND=1
      IND1=IND1+1
3     WRITE (6,16) I,PHI(I),IND1
4     CONTINUE
      IF (NPC.EQ.0) GO TO 6
      II=NVC
      DO 5 I=1,NPC
      II=II+1
      WRITE (6,17) II,PHI(II),I
      II=II+1
      WRITE (6,18) II,PHI(II),I
5     CONTINUE
6     CONTINUE
      IF (NEC.EQ.0) GO TO 8
      DO 7 I=1,NEC
      II=II+1
      WRITE (6,19) II,PHI(II),I
7     CONTINUE
8     CONTINUE
9     WRITE (6,21) IFNT
      RETURN
C
C
C
```

```
10    FORMAT (1H-,10X,35HBEST DISCRETE SOLUTION FOUND SO FAR,/)
11    FORMAT (30X,3HF =,E16.6//)
12    FORMAT (1H0,13X,31HOPTIMUM DISCRETE SOLUTION FOUND,/)
13    FORMAT (21X,12HMINIMUM  F =,E16.8//)
14    FORMAT (8X,*X(*,I2,*)=*,E16.8)
15    FORMAT (1H0,7X,*INEQUALITY CONSTRAINTS*,5X,*OCCURRING AT*,/)
16    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*SAMPLE POINT*,I2)
17    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*LOWER BOUND*,I3)
18    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*UPPER BOUND*,I3)
19    FORMAT (8X,*G(*,I2,*)=*,E16.8,5X,*EXTRA CONST*,I3)
20    FORMAT (1H1)
21    FORMAT (1H0,32HNUMBER OF FUNCTION EVALUATIONS =,I5)
      END
```

```
      SUBROUTINE DSPTL (NSP,NEC,KT,KP,NR,X,Z1,GRAD,SAMPT,G,GG,W1,IU)
      DIMENSION X(1), GRAD(1), SAMPT(3,1), G(1), GG(KP,1), W1(1), Z1(1),
     1 IU(1)
C
C     THIS SUBROUTINE PERFORMS CHECK OF THE GRADIENTS IN THE
C        USER SUPPLIED SUBROUTINES NETWRK AND USERCN
C     Z1 IS THE USER SUPPLIED VECTOR OF 2*KT ELEMENTS
C
      JJJ=0
      WRITE (6,12)
      DO 1 I=1,KT
      X(I)=Z1(I+KT)
    1 CONTINUE
      DO 5 IF=1,NSP
      CALL NETWRK (X,SAMPT(1,IF),F,GRAD,1)
      DO 3 I=1,KT
      IKT=I+KT
      Z=X(I)
      DX=1.E-4*X(I)
      IF (ABS(DX).LT.1.E-10) DX=1.E-10
      X(I)=Z+DX
      CALL NETWRK (X,SAMPT(1,IF),F2,GRAD,0)
      X(I)=Z-DX
      CALL NETWRK (X,SAMPT(1,IF),F1,GRAD,0)
      X(I)=Z
      Y=0.5*(F2-F1)/DX
      IF (ABS(Y).LT.1.E-14) Y=1.E-14
      IF (ABS(GRAD(I)).LT.1.E-14) GRAD(I)=1.E-14
      YPR=ABS((Y-GRAD(I))/Y)*100.0
      GG(I,IF)=GRAD(I)
      GG(IKT,IF)=Y
      IF (IF.NE.1) GO TO 2
      W1(I)=YPR
      IU(I)=1
      GO TO 3
    2 IF (YPR.LE.W1(I)) GO TO 3
      W1(I)=YPR
      IU(I)=IF
    3 CONTINUE
      IF (IF.NE.1) GO TO 5
      WRITE (6,19)
      DO 4 K=1,KT
      IKT=KT+K
      YY=1.-(GG(K,1)/GG(IKT,1))
      YYT=100.0*ABS(YY)
      WRITE (6,20) GG(K,1),GG(IKT,1),YYT
    4 CONTINUE
    5 CONTINUE
      WRITE (6,13)
      DO 6 I=1,KT
      IKT=I+KT
      IF=IU(I)
      YY=1.-(GG(I,IF)/GG(IKT,IF))
      W1(I)=100.0*ABS(YY)
      IF (ABS(W1(I)).GT.10.0) JJJ=1
      WRITE (6,15) GG(I,IF),GG(IKT,IF),W1(I),IF
    6 CONTINUE
```

```
      IF (NEC.EQ.0) GO TO 10
      WRITE (6,14)
      CALL USERCN (Z1,G,GG,NR,KP)
      DO 7 I=1,KP
      DO 7 J=1,NEC
      J1=J+NEC
7     GG(I,J1)=GG(I,J)
      DO 9 J=1,NEC
      J1=J+NEC
      DO 8 I=1,KP
      Z=Z1(I)
      DZ=1.E-4*Z1(I)
      IF (ABS(DZ).LT.1.E-10) DZ=1.E-10
      Z1(I)=Z+DZ
      CALL USERCN (Z1,G,GG,NR,KP)
      G2=G(J)
      Z1(I)=Z-DZ
      CALL USERCN (Z1,G,GG,NR,KP)
      G1=G(J)
      Z1(I)=Z
      Y=0.5*(G2-G1)/DZ
      IF (ABS(Y).LT.1.E-14) Y=1.E-14
      IF (ABS(GG(I,J1)).LT.1.E-14) GG(I,J1)=1.E-14
      YPR=ABS((Y-GG(I,J1))/Y)*100.0
      WRITE (6,15) GG(I,J1),Y,YPR,J
      IF (YPR.GT.10.0) JJJ=1
8     CONTINUE
      WRITE (6,16)
9     CONTINUE
10    IF (JJJ.NE.0) GO TO 11
      WRITE (6,17)
      RETURN
11    WRITE (6,18)
      CALL EXIT
C
C
C
12    FORMAT (1H1,*GRADIENT CHECK AT NOMINAL STARTING POINT*/1H ,40(*-*)
     1,/)
13    FORMAT (//1H0,*THE GRADIENTS FROM NETWRK HAVE BEEN CHECKED AT ALL
     1SAMPLE POINTS*//1H ,*THE LARGEST OVERALL DETECTED ERRORS ARE AS FO
     2LLOWS*///1H0,2X,*ANALYTICAL GRADIENTS*,2X,*NUMERICAL GRADIENTS*,2X
     3,*PERCENTAGE ERRORS*,2X,*SAMPLE POINT*)
14    FORMAT (///1H0,*THE GRADIENTS FROM THE USER SUPPLIED USERCN HAVE B
     1EEN CHECKED*//1H ,*FOR EACH GIVEN EXTRA CONSTRAINT THE ERRORS ARE
     2AS FOLLOWS*//1H0,2X,*ANALYTICAL GRADIENTS*,2X,*NUMERICAL GRADIENTS
     3*,2X,*PERCENTAGE ERRORS*,2X,*CONSTRAINT*)
15    FORMAT (1H0,5X,E14.6,7X,E14.6,5X,E14.6,9X,I3)
16    FORMAT (///)
17    FORMAT (1H0,///1H ,*GRADIENTS ARE O.K.*)
18    FORMAT (1H0,///1H ,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADI
     1ENTS ARE INCORRECT*/1H0,*PLEASE CHECK IT AGAIN*)
19    FORMAT (1H ,*THE GRADIENTS FROM THE USER SUPPLIED NETWRK HAVE BEEN
     1 CHECKED AT THE*//1H ,*FIRST SAMPLE POINT*///1H0,2X,*ANALYTICAL GR
     2ADIENTS*,2X,*NUMERICAL GRADIENTS*,2X,*PERCENTAGE ERRORS*)
20    FORMAT (1H0,5X,E14.6,7X,E14.6,5X,E14.6)
      END
```

```
      SUBROUTINE DSPIM (N,X,XE,IH,IK,RF,X1,JORDER)
C
      DIMENSION X(1), XE(N,IK,1), X1(1)
C
C     THIS SUBROUTINE EXTRAPOLATES ON THE VARIABLES TO ACCELERATE THE
C     CONVERGENCE IN ALGORITHM 3
C
C     A.V. FIACCO AND G.P. MCCORMICK, NONLINEAR PROGRAMMING- SEQUENTIAL
C     UNCONSTRAINED MINIMIZATION TECHNIQUES. NEW YORK- WILEY, 1968
C
      I=IH
      II=I+1
      DO 1 J=1,N
      XE(J,I,1)=X(J)
1     CONTINUE
      IF (I.LT.2) GO TO 11
      IF (I.GT.JORDER) GO TO 2
      IJ=I
      GO TO 3
2     IJ=JORDER+1
3     DO 5 L=2,IJ
      LL=L-1
      S=RF**LL
C
C     ESTIMATE OF THE ULTIMATE SOLUTION
C
      DO 4 J=1,N
      XE(J,I,L)=(S*XE(J,I,LL)-XE(J,I-1,LL))/(S-1.0)
4     CONTINUE
5     CONTINUE
      DO 6 J=1,N
      X1(J)=XE(J,I,IJ)
6     CONTINUE
      IF (I.EQ.IK) RETURN
C
C     ESTIMATE OF THE NEXT STARTING POINT
C
      DO 7 J=1,N
      XE(J,II,IJ)=XE(J,I,IJ)
7     CONTINUE
      DO 9 K=2,IJ
      L=IJ+1-K
      SS=RF**L
      DO 8 J=1,N
      XE(J,II,L)=((SS-1.)*XE(J,II,L+1)+XE(J,I,L))/SS
8     CONTINUE
9     CONTINUE
      DO 10 J=1,N
      X(J)=XE(J,II,1)
10    CONTINUE
      RETURN
11    DO 12 J=1,N
      X1(J)=XE(J,I,1)
12    CONTINUE
      RETURN
      END
```

SOC-105

TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING PART I - USER'S GUIDE, PART II - FORTRAN LISTING

J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu

September 1975, No. of Pages: Part I 47
Part II 27

Revised:

Key Words: Tolerancing, centering, continuous and discrete optimization, worst-case design

Abstract: This report describes the development, organization and implementation of a user-oriented computer program package called TOLOPT (TOLerance OPTimization), which can solve continuous and/or discrete worst-case tolerance assignment problems. Worst-case vertices can be automatically selected and optimization will lead to the most favorable nominal design simultaneously with the largest possible tolerances on specified toleranced components. The program contains recent techniques and algorithms for nonlinear programming. The optimization is carried out by subprograms substantially the same as ones in the DISOPT package. The full Fortran IV listing is included in this report as well as three circuit examples illustrating the use of and typical printouts from TOLOPT.

Description: Part I contains user's manual.
Part II contains Fortran listing.
Source deck available for $300.00.

Related Work: Represents further development of work reported in IEEE Trans. Microwave Theory and Techniques, vol. MTT-23, Aug. 1975, pp. 630-641. As for SOC-1.

Price: Part I $15.00.
Part II $85.00.