AD-A075 811 STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH F/G 9/2
DOCUMENTATION OF A COMPUTER PROGRAM FOR THE BOUND-AND-SCAN ALGO--ETC(U)
JUL 79 N F JACQMIN N00014-76-C-0418
UNCLASSIFIED TR-87 NL

1 OF 1
AD A075811



END
DATE FILMED
11-79
DDC

AD A075811

13

DOCUMENTATION OF A COMPUTER PROGRAM FOR THE BOUND-AND-SCAN ALGORITHM FOR INTEGER LINEAR PROGRAMMING

BY

NANCY E. JACQMIN

LEVEL II

TECHNICAL REPORT NO. 87
JULY 1979

PREPARED UNDER CONTRACT
N00014-76-C-0418 (NR-047-061)
FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

DDC
OCT 26 1979
E

See 1473 in back.

Reproduction in Whole or in Part is Permitted
for any Purpose of the United States Government

This document has been approved for public release
and sale; its distribution is unlimited.

DDC FILE COPY

DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

79 10 26 044

DOCUMENTATION OF A COMPUTER PROGRAM FOR THE BOUND-AND-SCAN ALGORITHM FOR INTEGER LINEAR PROGRAMMING

by

NANCY E. JACQMIN

TECHNICAL REPORT NO. 87

JULY 1979

PREPARED UNDER CONTRACT

NOOO14-76-C-0418 (NR-047-061)

FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

Reproduction in Whole or in Part is Permitted
for any Purpose of the United States Government

This document has been approved for public release
and sale; its distribution is unlimited.

This research was supported in part by National Science Foundation Grant NSF Number - MCS 76-81259 A01, Dept. of Operations Research Stanford University.

DEPARTMENT OF OPERATIONS RESEARCH

STANFORD UNIVERSITY

STANFORD, CALIFORNIA

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist. Avail and/or special
A

1. Introduction

The computer code BDSCAN, written in FORTRAN by Bruce H. Faaland to implement the bound-and-scan algorithm developed by Frederick S. Hillier [1], is documented and listed in this report. This program is designed to solve the pure integer linear programming problem,

$$\text{maximize} \quad x_0 = \sum_{j=1}^{n} c_j x_j$$

subject to

$$(1) \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1,2,..,m),$$

$$(2) \quad x_j \geq 0 \qquad (j = 1,2,..,n),$$

$$(3) \quad x_j \text{ is an integer} \qquad (j = 1,2,..,n).$$

This problem may also be written as maximize cx subject to $Ax \leq b$, $x \geq 0$, x integer where the constraint matrix A is $m \times n$, the cost coefficients c form a $1 \times n$ row vector, the right hand side b is an $m \times 1$ column vector and x is an $n \times 1$ column vector.

It is required that a suboptimal feasible (integer) solution has been obtained by a suboptimal algorithm such as that developed by Hillier [3] and documented in a companion report. See [2] for a detailed discussion of the bound-and-scan algorithm.

2. Main Program and Input Requirements

The main program for the bound-and-scan code BDSCAN serves to coordinate the 16 parts described in detail in [2] and summarized in [1]. The first card of input must contain NAME, any alphanumeric characters to identify the problem. The second card contains M, N and KL in 3I5 format. These variables have the following significance:

M = the number of rows in A.

N = the number of columns in A.

$$KL = \begin{cases} 1, & \text{if the optimal basis inverse is in normalized form} \\ 0, & \text{otherwise.} \end{cases}$$

This basis inverse corresponds to an optimal solution to the stated problem ignoring the integer requirements (the "LP problem"). It is in normalized form if the constraints $Ax \leq b$ were normalized as defined in [3, p. 605] before the LP problem was solved.

The third group of cards contains the arrays A, b and c in 15F5.0 format. The A matrix is read in one row at a time, then b is read in starting on a new card, followed by c also starting on a new card.

The fourth group of cards contains the basis inverse for an optimal solution to the LP problem; the format is 6F13.5. This matrix is read in row by row.

The fifth group of cards contains JPM(I), I = 1,..,M in 15I4 format where JPM(I) is the index of the I^{th} basic variable (including slacks) in the solution to the LP problem.

The sixth group of cards contains an optimal solution to the LP problem in 6F13.5 format. The final group of cards contains a suboptimal feasible solution to the ILP problem (the LP problem plus the integer requirements) in 6F13.5 format.

Another problem may then be input using the card sequence described above.

The input for a sample run of BDSCAN is given in Appendix II.

3. Restrictions Relevant to the Use of BDSCAN

As currently written the following restrictions apply to the input data and problem size for BDSCAN:

1. NAME must have $\leq$ 20 characters.
2. M must be $\leq$ 60.
3. N must be $\leq$ 60.
4. KL = 0 or 1.

To change the bounds on M and N you simply have to modify the dimension statements for A from A(61,61) to A(m' +1, n' + 1) where m' and n' are the new bounds on M and N respectively.

4. Description of Subprograms

Detailed comments on the specific function of each subroutine may be found in the listing of BDSCAN in Appendix I. Variables of note are defined in these comments as well. In general terms:

1. SUBROUTINE IOSUB -- Reads input cards and initiates the algorithm.
2. SUBROUTINES MATRIX, PERMUT -- Rearrange arrays according to parameter instructions.
3. SUBROUTINE FIRST1 -- Checks to see if the assumptions of the algorithms as specified in [1, pp. 643-644] are satisfied.
4. SUBROUTINE PART1 -- Finds the extreme points of the region defined in [1, pp. 649-650] using the procedure described there.
5. SUBROUTINE PART2 -- Finds upper bounds on the weights on these extreme points. [1, pp. 650-651]
6. SUBROUTINE PART3 -- Finds unconditional upper and lower integer bounds on the variables. [1, p. 651]
7. SUBROUTINE PART4 -- Partitions the variables into the four groups described in [1, pp. 651-652].
8. SUBROUTINE PART5 -- Obtains conditional upper and lower bounds on the variables in Groups 2, 3 and 4. [1, pp. 653-654]

9. SUBROUTINE PART6 -- Initialized the procedure for the Group 1 variables. [1, p. 653]

10. SUBROUTINE PART 7 -- Obtains a new solution for the Group 1 variables using the scheme described in [1, pp. 652-653].

11. SUBROUTINE PART8 -- Obtains an initial trial solution for the Group 2 variables. [1, p. 655]

12. SUBROUTINE PART9 -- Obtains a new trial solution for the Group 2 variables. [1, p. 655]

13. SUBROUTINE PART10 -- Determines if a given trial solution for the Group 2 variables is eligible using the method described in [1, pp. 655-658].

14. SUBROUTINE PART11 -- Initializes the bounding procedure for the variables in Groups 3 and 4. [1, pp. 658-661]

15. SUBROUTINE SOLVLP -- Initiates the solution of an LP via the simplex or dual simplex methods. These methods are described in [4].

16. SUBROUTINE SIMPLX -- Solves an LP via the simplex method.

17. SUBROUTINES DUAL, INPUT, DUSEX, MP2, MP3, MP5 -- Together solve an LP via the dual simplex method.

18. SUBROUTINE PART12 -- Obtains new lower and upper bounds for the variables in Groups 3 and 4. [1, pp. 660-661, 663]

19. SUBROUTINE PART13 -- Obtains an initial solution for the Group 3 variables. [1, p. 661]

20. SUBROUTINE PART14 -- Obtains a new solution for the Group 3 variables. [1, pp. 661-662]

21. SUBROUTINES PART15, SORT -- Together initialize the procedure for the Group 4 variables. [1, pp. 663-665]

22. SUBROUTINES PART16 -- Obtains all the feasible completions for the Group 4 variables. [1, pp. 663-664]

A brief outline of how the parts of the algorithm fit together appears in [1, pp. 665-667]. A step-by-step description of each part may be found in [2].

5. Description of Output

The following output is generated by BDSCAN. During execution of SUBROUTINE IOSUB the following are printed. The NAME of the problem is printed out followed by the dimension of the constraint matrix A. Then A, the right hand side b and the cost coefficient array c are printed in 15F8.2 format.

The indices of the variables that are basic in the LP solution are then printed followed by the LP solution itself, in 15F8.2 format. Next the suboptimal feasible (integer) solution that was input is printed. Finally, before execution of the actual algorithm is started, the number of non-slack variables that are basic in the LP solution is recorded [n_B in 1, p. 644].

If, during execution of SUBROUTINE FIRST1, a violation of the assumptions of the algorithm is found, the nature of the violation will be printed prior to program termination.

On return from SUBROUTINE PART1 the first n extreme points of the region described in [1, p. 646] are printed in 15F8.2 format, one point at a time.

Following SUBROUTINE PART2 the upper bounds on the weights on the first n extreme points are printed in 15F8.2 format.

On return from SUBROUTINE PART3 the unconditional lower and upper bounds (integer) on the variables are printed in 15F8.2 format. Then the product of the ranges of these bounds is printed in F30.0 format.

In SUBROUTINE PART7 a message is printed telling which component j of the vector X represents the variable through which eligible Group 1 solutions have been scanned (meaning x_j has reached its upper bound and been reset to 0 in the enumeration scheme described in [1, p. 653]). Each time j changes the message is printed. Also in this subroutine division by zero may result; if so,

just prior to termination the indices K and J are printed indicating that XX(K,J) = 0 is the reason for the error. (Note: reference to the J^{th} entry in the K^{th} extreme point printed after SUBROUTINE PART1 will confirm this.)

During execution of SUBROUTINE SIMPLX an unbounded variable may be found. If so this message will be printed on return to SUBROUTINE SOLVLP, the calling program.

If execution of the dual simplex method during SUBROUTINE PART11 finds that the problem being solved is infeasible, a message to that effect is printed on return to SUBROUTINE INPUT, the calling program.

In SUBROUTINE PART16 each time an improved feasible (integer) solution is found it is printed out in 15F8.2 format, followed by its objective function value.

At the end of execution of the algorithm the following information is printed out:

1. The final feasibility test slacks ($b - Ax^B$, where x^B is the optimal (integer) solution) in 15F8.2 format.
2. The number of times each part was entered.
3. The objective function value for the initial suboptimal feasible (integer) solution in F8.2 format.
4. The objective function value for the LP solution in F8.2 format.
5. The difference between the objective function value of the LP solution and that of the ILP solution, normalized (multiplied by $1/(\sum_{j=1}^{n} c_j^2)^{1/2}$), in F13.5 format.
6. The normalized difference between the objective function value of the ILP solution and that of the suboptimal integer solution in F13.5 format.
7. The ILP solution.
8. The objective function value of the ILP solution.

If more than one problem was included in the input data the information above will be printed out for each problem starting with its input NAME.

The output from a sample run of BDSCAN is given in Appendix III.

6. Further Comments

As currently written the FORTRAN code BDSCAN contains approximately 4300 source statements. All significant array storage is done in a common block called COMMON. When compiled on Stanford's IBM 370/168 under IBM's FORTRAN-H-extended compiler, with level two optimization, approximately 512K bytes of core are required for all instructions and data. The code is WATFIV compatible. BDSCAN was last revised in November 1978.

REFERENCES

1. Hillier, Frederick S., "A Bound-and-Scan Algorithm for Pure Integer Linear Programming with General Variables," Operations Research 17, 638-679 (1969).

2. _____, "An Optimal Bound-and-Scan Algorithm for Integer Linear Programming," Technical Report No. 13, Contract Nonr-225(89), Program in Operations Research, Stanford University, August 19, 1966.

3. _____, "Efficient Heuristic Procedures for Integer Linear Programming with an Interior," Operations Research 17, 600-637 (1969).

4. _____ and Gerald J. Lieberman, Introduction to Operations Research, Holden-Day, Inc., 1967.

APPENDIX I

LISTING OF COMPUTER PROGRAM BDSCAN

```
//NEJBDSC    JOB  NEJ$WB,'JACOMIN, N',REGION=512K,TIME=(1,0)
//*MAIN LINES=99
//   EXEC   FORTHCG,PARM.FORT='OPT=2'
//FORT.SYSIN DD *
C
C
C     ***********************************************************************
C     ***********************************************************************
C     **      THE PURPOSE OF THIS CODE IS TO SOLVE THE PURE INTEGER LINEAR  **
C     **      PROGRAMMING PROBLEM:                                          **
C     **                                                                    **
C     **                        MAXIMIZE  XZERO = CX                        **
C     **                  SUBJECT TO:                                       **
C     **                        AX . LE . B                                 **
C     **                         X . GE . 0                                 **
C     **                     ALL COMPONENTS OF X INTEGRAL                   **
C     **                                                                    **
C     **      WHERE:  THE DIMENSION OF A IS M BY N                          **
C     **              THE DIMENSION OF B IS M                               **
C     **              THE DIMENSION OF C IS N.                              **
C     ***********************************************************************
C     ***********************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      DIMENSION KALIAS(61),NBR(20)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,
     XLK,LL,LPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,
     XU,UEP,NOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,
     XSORT,UBAR,ULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,
     XXPRIME,XX,XPN2,XPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
  500 FORMAT(1H ,15F8.2)
  551 FORMAT('0',' THE NUMBER OF TIMES THROUGH PART',I2,' WAS',
     XT45,I8,'.')
  552 FORMAT('0',' THE FINAL INTEGER SOLUTION IS:')
  553 FORMAT('0',' THE OBJECTIVE FUNCTION VALUE IS',T45,F8.2,'.')
  310 FORMAT('1','THE FIRST N EXTREME POINTS OF THE N-SIMPLEX FORMED',
     X' BY THE N')
  400 FORMAT(' ','CONSTRAINTS THAT ARE BINDING ON THE LP SOLUTION PLUS',
     X' THE')
  401 FORMAT(' ','CONSTRAINT CX . GE . ZF ARE:')
  313 FORMAT('0','THE WEIGHTS ON THESE N EXTREME POINTS ARE BOUNDED ',
     X'ABOVE BY:')
  314 FORMAT('-','THE UNCONDITIONAL BOUNDS ON THE VARIABLES ARE:')
  315 FORMAT(' ',' LOWER BOUNDS:')
  316 FORMAT(' ',' UPPER BOUNDS:')
  319 FORMAT('0','THE PRODUCT OF THE UNCONDITIONAL RANGES IS',T45,F30.0)
  520 FORMAT('0',' THE FEASIBILITY TEST SLACKS ARE:')
  301 FORMAT('0',' THE OBJECTIVE FUNCTION VALUE FOR THE SUBOPTIMAL ',
     X'SOLUTION IS',T65,F8.2,'.')
  302 FORMAT('0',' THE OBJECTIVE FUNCTION VALUE FOR THE LP ',
     X'SOLUTION IS',T65,F8.2,'.')
  303 FORMAT('0',' THE NORMALIZED DIFFERENCE BETWEEN THE ',
```

```
      X'OPTIMAL LP AND ILP OBJECTIVE FUNCTION VALUES IS',T95,F13.5,'.')
304    FORMAT('0',' THE NORMALIZED DIFFERENCE BETWEEN THE ',
     X'OPTIMAL ILP AND THE SUBOPTIMAL OBJECTIVE FUNCTION ',
     X'VALUES IS',T95,F13.5,'.')
210    FORMAT('-','AT THE END OF THE ALGORITHM THE FOLLOWING ',
     X'INFORMATION IS KNOWN:')
2123   CONTINUE
       LARGE=-1
C
C      *****   NBR(I) COUNTS ENTRIES TO SUBROUTINE PARTI.   *****
C
       DO 570 I=1,20
570      NBR(I)=0
       JJ=0
       CALL IOSUB
C
C      *****   JJ=1000 INDICATES THERE IS NO MORE DATA TO BE RUN.   *****
C
       IF(JJ.EQ.1000) RETURN
       JJ=0
       CALL FIRST1
C
C      *****   JJ=10000 INDICATES SOME ASSUMPTION IS VIOLATED.   *****
C
       IF(JJ.EQ.10000) GO TO 2123
101    CONTINUE
       CALL PART1
       I=1
       NBR(I)=NBR(I)+1
       LA=N+1
C
C      *****   STORE THE (N+1)ST EXTREME POINT IN XX(N+1,-)  *****
C      *****   AND PRINT OUT THE FIRST N EXTREME POINTS      *****
C      *****   WITH THE COLUMNS IN JB ORDER.                 *****
C
       DO 12 J=1,N
         XX(N+1,J)=X(J)
12       CONTINUE
       WRITE(6,310)
       WRITE(6,400)
       WRITE(6,401)
       DO 311 I=1,N
         DO 312 J=1,N
           LL=JB(J)
312        TEMP(LL)=XX(I,J)
311      WRITE(6,500)(TEMP(J),J=1,N)
102    CONTINUE
       CALL PART2
       I=2
       NBR(I)=NBR(I)+1
C
C      *****   PRINT OUT THE UPPER BOUNDS ON THE    *****
C      *****   FIRST N EXTREME POINT WEIGHTS.       *****
C
       WRITE(6,313)
       WRITE(6,500)(Y(J),J=1,N)
103    CONTINUE
       CALL PART3
       I=3
       NBR(I)=NBR(I)+1
```

```
C
C     *****   PRINT OUT THE UNCONDITIONAL BOUNDS  *****
C     *****   ON THE VARIABLES IN JB ORDER.       *****
C
      WRITE(6,314)
      WRITE(6,315)
      DO 317 J=1,N
        LL=JB(J)
317     TEMP(LL)=AL(J)
      WRITE(6,500)(TEMP(J),J=1,N)
      WRITE(6,316)
      DO 318 J=1,N
        LL=JB(J)
318     TEMP(LL)=U(J)
      WRITE(6,500)(TEMP(J),J=1,N)
C
C     *****   PRINT OUT THE PRODUCT OF UNCONDITIONAL RANGES.   *****
C
      T=1.0
      DO 320 J=1,N
        IF((U(J)-AL(J)).LE.1.0) GO TO 320
        T=T*(U(J)-AL(J))
320     CONTINUE
      WRITE(6,319)T
104   CONTINUE
      CALL PART4
      I=4
      NBR(I)=NBR(I)+1
C
C     *****   N2=N MEANS THERE ARE NO VARIABLES IN GROUPS 3 AND 4.   *****
C
      IF(N2.EQ.N) GO TO 107
105   CONTINUE
      CALL PART11
      I=11
      NBR(I)=NBR(I)+1
C
C     *****   STORE THE ORDER OF THE VARIABLES INTO  *****
C     *****   GROUPS IN KPERM AND KALIAS.            *****
C
      DO 13 J=1,N
        KPERM(J)=JPERM(J)
        KALIAS(J)=JPERM(J)
13      CONTINUE
106   CONTINUE
      CALL PART15
      I=15
      NBR(I)=NBR(I)+1
C
C     *****   STORE THE COLUMN ORDER OF W IN JPM. STORE THE    *****
C     *****   ORDER OF THE VARIABLES INTO GROUPS IN JPERM.     *****
C
      DO 14 J=1,N
        JPM(J)=JPERM(J)
        JPERM(J)=KALIAS(J)
14      CONTINUE
107   CONTINUE
      CALL PART6
      I=6
      NBR(I)=NBR(I)+1
```

```
      C
      C     *****   STORE THE INITIAL INTEGER SOLUTION IN XB WITH THE      *****
      C     *****   VARIABLES IN GROUP ORDER.  STORE THE CORRESPONDING     *****
      C     *****   OBJECTIVE FUNCTION VALUE IN ZB.                       *****
      C
  108 DO 15 J=1,N
        LC=JPERM(J)
        XB(J)=XF(LC)
  15    CONTINUE
      ZB=ZF
      BETA=1.0
      DEL=1.0
      C
      C     *****   INITIALIZE PART 5.  SET THE WEIGHTS ON THE EXTREME    *****
      C     *****   POINTS.  RHO(I) IS THE WEIGHT ON EXTREME POINT I IN   *****
      C     *****   THE JB ORDERING SINCE LPERM IS THE INVERSE OF JPERM.  *****
      C
      RHP=(ZB-ZF)/(ZX-ZF)
      KK=N+1
      LL=NB+1
      DO 252 I=LL,N
        K=LPERM(I)
        RHO(I)=X(K)/XX(I,K)
  252   CONTINUE
      C
      C     *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN KPERM.    *****
      C
  109 DO 16 J=1,N
        KPERM(J)=JPERM(J)
  16    CONTINUE
      C
      C     *****   N1=N2 MEANS THERE ARE NO GROUP 2 VARIABLES.    *****
      C
      IF(N1.EQ.N2) GO TO 114
      KL=N1+1
      NN=N2
      CALL PART5
      I=5
      NBR(I)=NBR(I)+1
      C
      C     *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN JPERM.    *****
      C
      DO 17 J=1,N
        JPERM(J)=KPERM(J)
  17    CONTINUE
  950 CONTINUE
  110 CONTINUE
      CALL PART8
      I=8
      NBR(I)=NBR(I)+1
      C
      C     *****   SOLN=1.0 MEANS A TRIAL PARTIAL SOLUTION FOR THE    *****
      C     *****   GROUP 2 VARIABLES HAS BEEN FOUND.                 *****
      C
      IF(SOLN.EQ.1.0) GO TO 112
  111 CONTINUE
      CALL PART7(LARGE)
      I=7
      NBR(I)=NBR(I)+1
      C
```

```
      C      *****   NOSOLN=0 MEANS NO NEW ELIGIBLE SOLUTION    *****
      C      *****   EXISTS FOR THE GROUP 1 VARIABLES.          *****
      C
            IF(NOSOLN.EQ.0) GO TO 1000
            BETA=1.0
            DEL=1
            GO TO 109
      C
      C     *****   PUT XB INTO JB ORDER.   *****
      C
      1000  CALL PERMUT(XB,N,LPERM)
      C
      C     *****   COMPUTE AND PRINT OUT THE FINAL VALUE OF THE SLACKS.   *****
      C
            WRITE(6,210)
            DO 525 I=1,M
              T=0.0
              DO 526 J=1,N
      526       T=T+A(I,J)*XB(J)
      525     B(I)=B(I)-T
            WRITE(6,520)
            WRITE(6,500)(B(I),I=1,M)
            DO 571 I=1,16
              WRITE(6,551)I,NBR(I)
      571     CONTINUE
            T=0.0
            T=0.0
            DO 20 J=1,N
              TEMP(J)=XB(J)
              T=T+C(J)*XB(J)
      20      CONTINUE
            DO 21 J=1,N
              LL=JB(J)
              XB(LL)=TEMP(J)
      21      CONTINUE
            TT=0.0
      C
      C     *****   STORE THE OBJECTIVE FUNCTION NORMALIZATION FACTOR IN TT.   *****
      C
            DO 300 J=1,N
      300     TT=TT+C(J)*C(J)
            TT=SQRT(TT)
            T1=(ZX-T)/TT
            T2=(T-ZF)/TT
            WRITE(6,301)ZF
            WRITE(6,302)ZX
            WRITE(6,303)T1
            WRITE(6,304)T2
            WRITE(6,552)
      C
      C     *****   PRINT OUT THE OPTIMAL SOLUTION.   *****
      C
            WRITE(6,500)(XB(J),J=1,N)
            WRITE(6,553)T
            GO TO 2123
      112   CONTINUE
            CALL PART10
            I=10
            NBR(I)=NBR(I)+1
      951   CONTINUE
```

```
      C
      C     *****   SOLN=1.0 MEANS THE TRIAL SOLUTION FOR THE *****
      C     *****   GROUP 2 VARIABLES IS ELIGIBLE.            *****
      C
            IF(SOLN.EQ.1.0) GO TO 114
  113       CONTINUE
            CALL PART9
            I=9
            NBR(I)=NBR(I)+1
      C
      C     *****   SOLN=1.0 MEANS A NEW TRIAL PARTIAL SOLUTION   *****
      C     *****   FOR THE GROUP 2 VARIABLES HAS BEEN FOUND.     *****
      C
            IF(SOLN.EQ.1.0) GO TO 112
            GO TO 111
  114       IF(N2.EQ.N) GO TO 601
            IF(BETA.EQ.0.0) GO TO 116
            BETA=0.0
      C
      C     *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN KPERM.   *****
      C
  115       DO 22 J=1,N
              KPERM(J)=JPERM(J)
   22         CONTINUE
            KL=N2+1
            NN=N3
            CALL PART5
            I=5
            NBR(I)=NBR(I)+1
      C
      C     *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN JPERM.   *****
      C
            DO 23 J=1,N
              JPERM(J)=KPERM(J)
   23         CONTINUE
  952       CONTINUE
  116       CONTINUE
            CALL PART13
            I=13
            NBR(I)=NBR(I)+1
      C
      C     *****   SOLN=0.0 MEANS NO INITIAL PARTIAL SOLUTION   *****
      C     *****   FOR THE GROUP 3 VARIABLES EXISTS.           *****
      C
            IF(SOLN.EQ.0.0.AND.N1.EQ.N2) GO TO 111
            IF(SOLN.EQ.0.0) GO TO 113
  117       IF(DEL.EQ.0.0) GO TO 119
            DEL=0.0
      C
      C     *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN KPERM.   *****
      C
  118       DO 25 J=1,N
              KPERM(J)=JPERM(J)
   25         CONTINUE
            KL=N3+1
            NN=N
            CALL PART5
            I=5
            NBR(I)=NBR(I)+1
      C
```

```
C      *****   STORE THE ORDER OF THE VARIABLES INTO GROUPS IN JPERM.    *****
C
       DO 26 J=1,N
         JPERM(J)=KPERM(J)
26       CONTINUE
953    CONTINUE
119    KL=N3+1
       NN=N
       DO 9832 KM=KL,NN
         J=KM
         CALL PART12
9832     CONTINUE
       I=12
       NBR(I)=NBR(I)+1
954    CONTINUE
120    CONTINUE
       CALL PART16
       I=16
       NBR(I)=NBR(I)+1
121    CONTINUE
       CALL PART14
       I=14
       NBR(I)=NBR(I)+1
C
C      *****   SOLN=0.0 MEANS NO ELIGIBLE PARTIAL SOLUTION   *****
C      *****   EXISTS FOR THE GROUP 3 VARIABLES.             *****
C
       IF(SOLN.EQ.0.0.AND.N1.EQ.N2) GO TO 111
       IF(SOLN.EQ.0.0) GO TO 113
       GO TO 119
601    CONTINUE
C
C      *****   CHECK THE SIGN OF THE CURRENT FEASIBILITY TEST SLACKS.    *****
C
       DO 602 I=1,M
         T=0.0
         DO 603 J=1,N
           JJ=JPERM(J)
603        T=T+A(I,JJ)*X(J)
         T=T-B(I)
         IF(T.GT.0.0) GO TO 113
602      CONTINUE
       T=0.0
C
C      *****   IF THE SLACKS ARE NON-NEGATIVE COMPUTE THE    *****
C      *****   OBJECTIVE FUNCTION VALUE.                     *****
C
       DO 604 J=1,N
         JJ=JPERM(J)
604      T=T+C(JJ)*X(J)
       IF(T.LE.ZB) GO TO 113
C
C      *****   REPLACE ZB AND XB WITH THE NEW SOLUTION.   *****
C
       ZB=T
       DO 605 J=1,N
605      XB(J)=X(J)
       GO TO 113
       END
C
```

```
      C
      C
      C
      C
            SUBROUTINE IDSUB
      C
      C
      C     ****************************************************************
      C     *      THIS SUBROUTINE GATHERS INFORMATION TO BE USED BY THE ALGORITHM     *
      C     *      AS IT IS EXECUTED.                                        *
      C     ****************************************************************
      C
      C
            DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
           1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
           2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
           3NAME(20),RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
           4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),
           5XX(61,61),XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
            COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
           XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
           XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
           XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
           XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
           XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      200   FORMAT(15F5.0)
      201   FORMAT('0','THE CONSTRAINT MATRIX A IS',I3,'  BY',I3,'.')
      203   FORMAT(1H ,15I4)
      574   FORMAT(20A4)
      575   FORMAT(1H1,5X,20A4)
      500   FORMAT(1H ,15F8.2)
      503   FORMAT('0','THE CONSTRAINT MATRIX A(I,J) IS:')
      504   FORMAT('0','THE RIGHT HAND SIDE B(I) IS:')
      505   FORMAT('0','THE COST COEFFICIENTS C(J) ARE:')
      511   FORMAT('0','THE LP SOLUTION IS:')
      512   FORMAT('0','THE SUBOPTIMAL FEASIBLE SOLUTION IS:')
      5001  FORMAT(6F13.5)
      5711  FORMAT(15I4)
      550   FORMAT('0','THE NUMBER OF VARIABLES BASIC IN THE LP SOLUTION IS',
           XI3,'.')
      204   FORMAT('0','FOLLOWING, IN ORDER OF INCREASING I, IS THE ITH ',
           X'BASIC VARIABLE, INCLUDING SLACKS:')
      205   FORMAT(3I5)
      C
      C     *****   READ IN DATA.   *****
      C
            READ(5,574,END=2000)(NAME(I),I=1,20)
            WRITE(6,575)(NAME(I),I=1,20)
      C
      C     *****   KL=1 MEANS BASIS INVERSE HAS BEEN NORMALIZED.   *****
      C
            READ(5,205) M,N,KL
            WRITE(6,201)M,N
            DO 183 I=1,M
      183     READ(5,200)(A(I,J),J=1,N)
            READ(5,200)(B(I),I=1,M)
            READ(5,200)(C(J),J=1,N)
            WRITE(6,503)
            DO 182 I=1,M
      182     WRITE(6,500)(A(I,J),J=1,N)
```

```
      WRITE(6,504)
      WRITE(6,500)(B(I),I=1,M)
      WRITE(6,505)
      WRITE(6,500)(C(J),J=1,N)
C
C*****   BB CONTAINS THE BASIS INVERSE FOR THE   *****
C*****   OPTIMAL LP SOLUTION.                    *****
C
      READ(5,5001)((BB(I,J),J=1,M),I=1,M)
C
C*****   JPM(I) IS THE ITH BASIC VARIABLE, INCLUDING SLACKS.   *****
C
      READ(5,5711)(JPM(I),I=1,M)
      WRITE(6,204)
      WRITE(6,203)(JPM(I),I=1,M)
C
C*****   X CONTAINS THE OPTIMAL LP SOLUTION.   *****
C
      READ(5,5001)(X(J),J=1,N)
C
C*****   XF CONTAINS THE INITITAL FEASIBLE INTEGER SOLUTION.   *****
C
      READ(5,5001)(XF(J),J=1,N)
      WRITE(6,511)
      WRITE(6,500)(X(I),I=1,N)
      WRITE(6,512)
      WRITE(6,500)(XF(I),I=1,N)
      IF(KL.NE.1)GO TO 230
C
C*****   UN-NORMALIZE THE BASIS INVERS.  THE ITH COLUMN OF     *****
C*****   THE BASIS INVERSE CORRESPONDS TO THE ITH ROW OF THE   *****
C*****   CONSTRAINT MATRIX A, SO IF THE ITH ROW OF A WERE      *****
C*****   MULTIPLIED BY SOME NUMBER SUM WHEN NORMALIZING THE    *****
C*****   CONSTRAINTS, THE UN-NORMALIZED INVERSE WOULD BE       *****
C*****   OBTAINED BY DIVIDING THE ITH COLUMN OF THE INVERSE    *****
C*****   BY SUM.                                               *****
C
      DO 231 I=1,M
        SUM=0.0
        DO 232 J=1,N
  232     SUM=SUM+A(I,J)*A(I,J)
        SUM=SQRT(SUM)
        DO 233 J=1,M
  233     BB(J,I)=BB(J,I)/SUM
  231   CONTINUE
  230 CONTINUE
C
C*****   IDENTIFY JB(1),...,JB(N) AS SUBSCRIPTS OF THE LP    *****
C*****   VARIABLES SO THAT JB(1),...,JB(NB) ARE BASIC IN     *****
C*****   THE LP SOLUTION.  IDENTIFY KN(I), I=1,...,M SO      *****
C*****   THAT THE SUBSCRIPTS N+KB(I), I=1,...,NB CORRESPOND *****
C*****   TO THE NON-BASIC SLACK VARIABLES IN THE LP          *****
C*****   SOLUTION.                                           *****
C
      NB=0
      DO 234 I=1,M
  234   IF(JPM(I).LE.N)NB=NB+1
      WRITE(6,550)NB
      N1=N-NB
      DO 241 J=1,N
```

```
      241     LPERM(J)=0
              DO 242 I=1,M
      242     IPERM(I)=0
              DO 243 I=1,M
                LL=JPM(I)
                IF(LL.GT.N) GO TO 244
                LPERM(LL)=I
                GO TO 243
      244       LL=LL-N
                IPERM(LL)=I
      243       CONTINUE
              L1=0
              L2=NB
              DO 245 J=1,N
                IF(LPERM(J).EQ.0)GO TO 246
                L1=L1+1
                JB(L1)=J
                GO TO 245
      246       L2=L2+1
                JB(L2)=J
      245       CONTINUE
              L1=0
              L2=NB
              DO 247 I=1,M
                IF(IPERM(I).EQ.0) GO TO 248
                L2=L2+1
                KN(L2)=I
                GO TO 247
      248       L1=L1+1
                KN(L1)=I
      247       CONTINUE
      C
      C       *****   KPERM PUTS THE BASIC VARIABLES IN THE LP SOLUTION IN *****
      C       *****   NUMERICAL ORDER.  JPM IS REDEFINED TO DO THE SAME.   *****
      C
              DO 250 K=1,M
      250       KPERM(K)=K
              MM=M-1
              DO 251 K=1,MM
                LT=JPM(K)
                KK=K+1
                DO 252 J=KK,M
                  IF(JPM(J).GE.LT) GO TO 252
                  LT=JPM(J)
                  JPM(J)=JPM(K)
                  JPM(K)=LT
                  LHOLD=KPERM(J)
                  KPERM(J)=KPERM(K)
                  KPERM(K)=LHOLD
      252         CONTINUE
      251       CONTINUE
      C
      C       *****   REORDER THE INVERSE ROWS TO CORRESPOND TO KPERM.   *****
      C
              LM=2
              DO 253 J=1,M
      253       CALL MATRIX(BB,M,KPERM,J,LM)
      C
      C       *****   REORDER THE COLUMNS OF A ACCORDING TO JB SO THAT THE   *****
      C       *****   BASIC VARIABLES ARE FIRST.  REORDER THE COLUMNS OF BB  *****
```

```
C      *****   ACCORDING TO KN SO THAT THE BINDING CONSTRAINTS ARE     *****
C      *****   FIRST.                                                  *****
C
       LM=1
       DO 254 I=1,M
         CALL MATRIX(A,N,JB,I,LM)
254    CALL MATRIX(BB,M,KN,I,LM)
C
C      *****   REORDER THE ROWS OF A ACCORDING TO KN SO THAT    *****
C      *****   THE BINDING CONSTRAINTS ARE FIRST.               *****
C
       LM=2
       DO 255 J=1,N
255    CALL MATRIX(A,M,KN,J,LM)
C
C      *****   REORDER XF,X AND C ACCORDING TO JB SO THAT THE *****
C      *****   BASIC VARIABLES ARE FIRST.  REORDER B ACCORD-  *****
C      *****   ING TO KN SO THAT THE BINDING CONSTRAINTS ARE  *****
C      *****   FIRST.                                        *****
C
       CALL PERMUT(XF,N,JB)
       CALL PERMUT(X,N,JB)
       CALL PERMUT(C,N,JB)
       CALL PERMUT(B,M,KN)
       RETURN
2000   JJ=1000
       RETURN
       END
C
C
C
C
       SUBROUTINE MATRIX(BB,LR,JPERM,I,LM)
C
C      ***********************************************************************
C      *      THIS SUBROUTINE REORDERS THE ROWS (LM=2) OR COLUMNS (LM=1) OF  *
C      *      BB ACCORDING TO JPERM.                                         *
C      ***********************************************************************
C
C
       DIMENSION BB(61,61),TEMP(61),JPERM(61)
       IF(LM.EQ.2) GO TO 30
       DO 10 J=1,LR
         TEMP(J)=BB(I,J)
10       CONTINUE
       DO 11 J=1,LR
         LL=JPERM(J)
         BB(I,J)=TEMP(LL)
11       CONTINUE
       GO TO 15
30     CONTINUE
       DO 12 J=1,LR
         TEMP(J)=BB(J,I)
12       CONTINUE
       DO 13 J=1,LR
         LL=JPERM(J)
         BB(J,I)=TEMP(LL)
13       CONTINUE
```

```
      15    RETURN
            END
      C
      C
            SUBROUTINE PERMUT(Y,LR,JPERM)
      C
      C     ****************************************************************
      C     *      THIS SUBROUTINE PUTS Y(1),....,Y(LR) IN THE JPERM ORDERING.*
      C     ****************************************************************
      C
      C
            DIMENSION Y(61),TEMP(61),JPERM(61)
            DO 10 J=1,LR
              TEMP(J)=Y(J)
      10      CONTINUE
            DO 11 J=1,LR
              LL=JPERM(J)
              Y(J)=TEMP(LL)
      11      CONTINUE
            RETURN
            END
      C
      C     ****************************************************************
      C     ****************************************************************
      C
            SUBROUTINE FIRST1
      C     ****************************************************************
      C     ****************************************************************
      C     *      THIS SUBROUTINE CHECKS TO SEE IF THE ASSUMPTIONS OF THE *
      C     *      ALGORITHM HOLD.                                         *
      C     ****************************************************************
      C     ****************************************************************
      C
            DIMENSION A(51,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
           1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
           2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
           3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
           4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),XX(61,61),XPRIME(61),
           6Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61),VV(61),XB(61)
            COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
           XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,
           XLK,LL,LPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,
           XU,UEP,NOSOLN,NZERP,RHC,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,
           XSORT,LBAR,ULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,
           XXPRIME,XX,XPN2,XPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      400   FORMAT('-','LP SOLUTION IS INTEGER.')
      401   FORMAT('-','SUBOPTIMAL SOLUTION IS OPTIMAL.')
      402   FORMAT('-','LP SOLUTION IS NOT UNIQUE.')
      C
      C     *****   STORE THE OBJECTIVE FUNCTION VALUE OF THE  *****
      C     *****   LP SOLUTION IN ZX.                         *****
      C
            ZX=0.0
            DO 1 J=1,N
      1       ZX=ZX+C(J)*X(J)
      C
      C     *****   CHECK TO SEE IF THE LP SOLUTION IS INTEGER.   *****
```

```
      C
      DO 2 J=1,N
        IF(X(J).NE.AINT(X(J))) GO TO 3
    2   CONTINUE
      WRITE(6,400)
      JJ=10000
      RETURN
    3 CONTINUE
      C
      C     *****   STORE THE OBJECTIVE FUNCTION VALUE OF THE INITIAL    *****
      C     *****   INTEGER SOLUTION IN ZF.                              *****
      C
      ZF=0.0
      DO 4 J=1,N
    4   ZF=ZF+C(J)*XF(J)
      C
      C     *****   CHECK XF FOR OPTIMALITY.    *****
      C
      IF(ZF.LT.(ZX-.001)) GO TO 5
      WRITE(6,401)
    6 JJ=10000
      RETURN
    5 CONTINUE
      LL=NB+1
      C
      C     *****   CALCULATE THE SIMPLEX MULTIPLIERS.    *****
      C
      DO 8 I=1,M
        T=0.0
        DO 9 J=1,NB
    9     T=T+C(J)*BB(J,I)
        TEMP(I)=T
    8   CONTINUE
      C
      C     *****   CHECK TO SEE IF THE LP SOLUTION IS UNIQUE.    *****
      C
      IF(LL.GT.N) GO TO 13
      DO 10 J=LL,N
        T=0.0
        DO 11 I=1,M
   11     T=T+TEMP(I)*A(I,J)
        IF(T.EQ.C(J)) GO TO 12
   10   CONTINUE
      GO TO 13
   12 WRITE(6,402)
      GO TO 6
   13 DO 14 I=1,NB
        IF(TEMP(I).EQ.0.0) GO TO 12
   14   CONTINUE
      RETURN
      END
      C
      C
      C
      C
      C
      SUBROUTINE PART1
      C
      C
      ***********************************************************************
```

```
C      *      FIND THE (N+1) EXTREME POINTS OF THE N-SIMPLEX FORMED BY THE N      *
C      *      CONSTRAINTS THAT ARE BINDING ON THE LP SOLUTION PLUS THE NEW        *
C      *      CONSTRAINT CX . GE . ZF.                                            *
C      ************************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),
     5XX(61,61),XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
C
C      *****   INTRODUCE THE NEW CONSTRAINT CX . GE . ZF AND    *****
C      *****   INTRODUCE X(N+M+1) AS THE ADDITIONAL BASIC       *****
C      *****   VARIABLE.                                        *****
C
      LL=NB+1
      KK=M+1
      DO 71 J=1,M
        T=0.0
        DO 72 K=1,NB
          LK=K
          T=T+C(LK)*BB(K,J)
72        CONTINUE
        BB(KK,J)=T
        BB(J,KK)=0.0
71      CONTINUE
      BB(KK,KK)=1.0
      L=M+N+1
      X(L)=ZX-ZF
C
C      *****   INTRODUCE X(N+I) INTO THE BASIS AND REMOVE THE   *****
C      *****   VARIABLE INTRODUCED INTO THE PRECEDING BASIS.    *****
C      *****   THE FIRST N COMPONENTS OF THE NEW BASIC SOLUTION *****
C      *****   FORM XX(I,-), THE ITH EXTREME POINT.             *****
C
      DO 73 I=1,NB
        T=1.0/BB(KK,I)
        DO 74 J=1,M
          Y(J)=-BB(J,I)*T
74        CONTINUE
        Y(KK)=T
        IF(I.GT.1)GO TO 75
        DO 76 J=1,NB
          LK=J
          XX(I,LK)=X(LK)+Y(J)*X(L)
76        CONTINUE
        GO TO 78
75      DO 77 J=1,NB
          LK=J
          XX(I,LK)=XX(I-1,LK)+Y(J)*X(L)
77        CONTINUE
78      X(L)=X(L)*Y(KK)
```

```
      IF(LL.GT.N) GO TO 100
      DO 79 J=LL,N
        LK=J
        XX(I,LK)=0.0
   79   CONTINUE
  100 CONTINUE
C
C     *****   UPDATE THE INVERSE FOR THE NEW BASIS.   *****
C
      Y(KK)=Y(KK)-1.0
      DO 80 K=1,KK
        DO 81 J=1,KK
          BB(K,J)=BB(K,J)+Y(K)*BB(KK,J)
   81     CONTINUE
   80   CONTINUE
   73 CONTINUE
      IF(LL.GT.N) GO TO 101
C
C     *****   INTRODUCE X(I) INTO THE BASIS AND REMOVE THE   *****
C     *****   VARIABLE INTRODUCED INTO THE PRECEDING BASIS.  *****
C     *****   THE FIRST N COMPONENTS OF THE NEW BASIC SOLU-  *****
C     *****   TION FORM XX(I,-), THE ITH EXTREME POINT.      *****
C
      DO 85 I=LL,N
        T=0.0
        LK=I
        DO 86 J=1,M
          T=T+BB(KK,J)*A(J,LK)
   86     CONTINUE
        T=T-BB(KK,KK)*C(LK)
        T=1.0/T
        Y(KK)=T
        DO 87 K=1,M
          TT=0.0
          DO 88 J=1,M
            TT=TT+BB(K,J)*A(J,LK)
   88       CONTINUE
          TT=TT-BB(K,KK)*C(LK)
          Y(K)=-TT*T
   87     CONTINUE
        DO 90 J=1,NB
          LM=J
          XX(I,LM)=XX(I-1,LM)+Y(J)*X(L)
   90     CONTINUE
        DO 91 J=LL,N
          LM=J
          XX(I,LM)=0.0
   91     CONTINUE
        X(L)=X(L)*Y(KK)
        XX(I,LK)=X(L)
C
C     *****   UPDATE THE INVERSE FOR THE NEW BASIS.   *****
C
        Y(KK)=Y(KK)-1.0
        DO 92 K=1,KK
          DO 93 J=1,KK
            BB(K,J)=BB(K,J)+Y(K)*BB(KK,J)
   93       CONTINUE
   92     CONTINUE
   85   CONTINUE
```

```
  101   CONTINUE
        RETURN
        END
C
C
C
C
C
      SUBROUTINE PART2
C
C
C     ****************************************************************************
C     *      THIS SUBROUTINE OBTAINS UPPER BOUNDS ON THE MAXIMUM VALUE OF THE    *
C     *      WEIGHTS ON EACH EXTREME POINT BASED ON THE NONNEGATIVITY CON-       *
C     *      STRAINTS THAT WERE NOT BINDING ON THE LP SOLUTION.                  *
C     ****************************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61,61),VV(61),XB(61),XX(61,61),
     5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
C
C     *****   INITIALIZE THE BOUNDS.  STORE THE BOUND ON THE      *****
C     *****   WEIGHT ON EXTREME POINT I IN Y(I).                  *****
C
      DO 101 J=1,N
        Y(J)=1.0
  101   CONTINUE
      Y(N+1)=1.0
C
C     *****   VARIABLE NQ COUNTS THE NUMBER OF COLUMNS IN THE FIRST N    *****
C     *****   EXTREME POINTS THAT HAVE NEGATIVE ENTRIES.  INDEXQ(I)=J    *****
C     *****   MEANS THERE IS A NEGATIVE ENTRY IN COLUMN J OF SOME        *****
C     *****   EXTREME POINT.                                             *****
C
      NQ=0
      DO 102 J=1,NB
        DO 103 I=1,N
          IF(XX(I,J).GE.0.0) GO TO 103
          NQ=NQ+1
          DO 106 K=1,N
            BB(J,K)=0.0
  106       CONTINUE
          INDEXQ(NQ)=J
          DELTA(J)=1.0
          GO TO 102
  103     CONTINUE
  102   CONTINUE
      IF(NQ.EQ.0)RETURN
      IF(NB.EQ.N) GO TO 11
      LL=NB+1
```

```
      DO 104 I=LL,N
        T=XX(I,I)
        Y(I)=AINT(T)/T
  104   CONTINUE
   11 CONTINUE
      L=1
      J=INDEXO(1)
      LK=N+1
      DO 10 K=1,N
   10   XX(LK,K)=X(K)
      GO TO 121
  111 IF(DELTA(J))121,190,121
  121 DO 122 K=1,N
        JPERM(K)=K
  122   CONTINUE
C
C     *****   JPERM ORDERS THE ENTRIES IN COLUMN J OF THE EXTREME   *****
C     *****   POINTS, LARGEST ENTRIES FIRST.                        *****
C
      JPERM(LK)=LK
      DO 123 K=1,LK
        KK=JPERM(K)
        MAX=XX(KK,J)
        DO 124 I=K,LK
          JJ=JPERM(I)
          IF(MAX.GE.XX(JJ,J)) GO TO 124
          MAX=XX(JJ,J)
          JPERM(I)=JPERM(K)
          JPERM(K)=JJ
  124     CONTINUE
  123   CONTINUE
C
C     *****   SEARCH FOR A NEGATIVE ENTRY IN COLUMN J.   *****
C
      I=1
  131 IF(XX(I,J))132,135,135
  132 KL=1
      SUM=0.0
      SUM1=0.0
      SK1=XX(I,J)
      GO TO 141
  135 IF(I.EQ.N) GO TO 180
      I=I+1
      GO TO 131
C
C     *****   SK .GE. 0 SIGNALS THAT THE LARGEST FEASIBLE WEIGHT ON   *****
C     *****   XX(I,J) HAS BEEN REACHED.  STORE THE NEW BOUND IN Y(I).   *****
C
  141 KK=JPERM(KL)
      SUM=SUM+Y(KK)
      IF(SUM.GT.1.0) GO TO 156
      SUM1=SUM1+Y(KK)*XX(KK,J)
      SK=SUM1+(1.0-SUM)*XX(I,J)
  151 IF(SK.GE.0.0) GO TO 156
      KL=KL+1
      SK1=SK
      GO TO 141
  156 KK=JPERM(KL)
      THETA=SK1/(XX(I,J)-XX(KK,J))
      Y(I)=1.0-(SUM+THETA)+Y(KK)
```

```
1020.                IF(KL.EQ.1)GO TO 12
1021.                LKL=KL-1
1022.                DO 157 K=1,LKL
1023.                  KK=JPERM(K)
1024.                  IF(KK.NE.LK)BB(J,KK)=Y(KK)
1025.          157     CONTINUE
1026.          12    CONTINUE
1027.                KK=JPERM(KL)
1028.                IF(KK.NE.LK)BB(J,KK)=THETA
1029.                GO TO 135
1030.          180   IF(L.EQ.NQ)GO TO 185
1031.                L=L+1
1032.                J=INDEXQ(L)
1033.                GO TO 111
1034.          C
1035.          C     *****   CHECK FOR POSSIBLE FURTHER TIGHTENING OF THE    *****
1036.          C     *****   BOUNDS.  DELTA(KK)=1 MEANS FURTHER CHANGES      *****
1037.          C     *****   SHOULD BE INVESTIGATED.                         *****
1038.          C
1039.          185   T=0.0
1040.                DO 186 K=1,NQ
1041.                  KK=INDEXQ(K)
1042.                  DELTA(KK)=0.0
1043.                  DO 187 JJ=1,N
1044.                    IF(Y(JJ).GE.BB(KK,JJ))GO TO 187
1045.                    DELTA(KK)=1.0
1046.                    T=1.0
1047.                    GO TO 186
1048.          187       CONTINUE
1049.          186     CONTINUE
1050.                IF(T.EQ.0.0)RETURN
1051.                L=1
1052.                J=INDEXQ(1)
1053.                GO TO 111
1054.                END
1055.          C
1056.          C
1057.          C
1058.          C
1059.          C
1060.                SUBROUTINE PART3
1061.          C
1062.          C
1063.          C     ***************************************************************
1064.          C     *      USE THE (N+1) EXTREME POINTS AND THE WEIGHTS Y TO FIND UPPER AND    *
1065.          C     *      LOWER INTEGER BOUNDS ON THE VALUES OF THE VARIABLES.     *
1066.          C     ***************************************************************
1067.          C
1068.          C
1069.                DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
1070.               1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
1071.               2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
1072.               3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
1073.               4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
1074.               5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
1075.                COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
1076.               XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
1077.               XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
1078.               XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
1079.               XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
```

```
      XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
C
C     *****   JPERM ORDERS THE ENTRIES IN COLUMN J OF THE EXTREME        *****
C     *****   POINTS, LARGEST ENTRIES FIRST.  FIND AN UPPER BOUND ON     *****
C     *****   X(J) BY PUTTING MAXIMAL WEIGHT ON THE LARGEST COLUMN J     *****
C     *****   ENTRIES UNTIL THE WEIGHTS SUM TO 1, STORE THIS VALUE IN    *****
C     *****   U(J).                                                      *****
C
      LK=N+1
      DO 202 J=1,N
        DO 201 K=1,LK
          JPERM(K)=K
  201     CONTINUE
        SUM=0.0
        THETA=0.0
        DO 203 K=1,LK
          KK=JPERM(K)
          MAX=XX(KK,J)
          DO 204 I=K,LK
            JJ=JPERM(I)
            IF(MAX.GE.XX(JJ,J))GO TO 204
            MAX=XX(JJ,J)
            JPERM(I)=JPERM(K)
            JPERM(K)=JJ
  204       CONTINUE
          LL=JPERM(K)
          SUM=SUM+Y(LL)
          IF(SUM-1.0)205,206,207
  205     THETA=THETA+Y(LL)*XX(LL,J)
          GO TO 203
  206     THETA=THETA+Y(LL)*XX(LL,J)
          GO TO 208
  207     SUM=SUM-Y(LL)
          THETA=THETA+(1.0-SUM)*XX(LL,J)
          GO TO 208
  203     CONTINUE
  208   U(J)=AINT(THETA+.01)
  202   CONTINUE
      IF(NB.EQ.N)GO TO 13
      LK=NB+1
C
C     *****   ASSIGN A LOWER BOUND OF 0.0 TO NONBASIC      *****
C     *****   VARIABLES X(J), STORE IT IN AL(J).           *****
C
      DO 209 J=LK,N
        AL(J)=0.0
  209   CONTINUE
   13 CONTINUE
      LK=N+1
C
C     *****   JPERM ORDERS THE ENTRIES IN COLUMN J OF THE EXTREME        *****
C     *****   POINTS, SMALLEST ENTRIES FIRST.  FIND A LOWER BOUND ON     *****
C     *****   X(J) BY PUTTING MAXIMAL WEIGHT ON THE SMALLEST COLUMN J    *****
C     *****   ENTRIES UNTIL THE WEIGHTS SUM TO 1, STORE THIS VALUE IN    *****
C     *****   AL(J).                                                     *****
C
      DO 212 J=1,NB
        DO 211 K=1,LK
          JPERM(K)=K
  211     CONTINUE
```

```
1140.                SUM=0.0
1141.                THETA=0.0
1142.                DO 213 K=1,LK
1143.                  KK=JPERM(K)
1144.                  MIN=XX(KK,J)
1145.                  DO 214 I=K,LK
1146.                    JJ=JPERM(I)
1147.                    IF(MIN.LE.XX(JJ,J))GO TO 214
1148.                    MIN=XX(JJ,J)
1149.                    JPERM(I)=JPERM(K)
1150.                    JPERM(K)=JJ
1151.        214       CONTINUE
1152.                  LL=JPERM(K)
1153.                  SUM=SUM+Y(LL)
1154.                  IF(SUM-1.0)215,216,217
1155.        215       THETA=THETA+Y(LL)*XX(LL,J)
1156.                  GO TO 213
1157.        216       THETA=THETA+Y(LL)*XX(LL,J)
1158.                  GO TO 219
1159.        217       SUM=SUM-Y(LL)
1160.                  THETA=THETA+(1.0-SUM)*XX(LL,J)
1161.                  GO TO 218
1162.        213       CONTINUE
1163.        218     IF(THETA)219,219,220
1164.        220     IF(THETA.EQ.AINT(THETA))GO TO 221
1165.                THETA=AINT(THETA)+1.0
1166.        221     AL(J)=THETA
1167.                GO TO 212
1168.        219     AL(J)=0.0
1169.        212     CONTINUE
1170.              RETURN
1171.              END
1172.        C
1173.        C
1174.        C
1175.        C
1176.        C
1177.              SUBROUTINE PART4
1178.        C
1179.        C
1180.        C     ****************************************************************
1181.        C     *      PARTITION THE VARIABLES INTO FOUR GROUPS TO BE TREATED DIFFERENTLY *
1182.        C     *      BY THE ALGORITHM.                                       *
1183.        C     ****************************************************************
1184.        C
1185.        C
1186.              DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
1187.             1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
1188.             2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
1189.             3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
1190.             4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
1191.             5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
1192.              COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
1193.             XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
1194.             XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
1195.             XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SQRT,UBAR,
1196.             XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
1197.             XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
1198.        C
1199.        C     *****    THE VARIABLES HAVE BEEN INDEXED SO THE BASIC VARIABLES    *****
```

```
1200.      C      *****   IN THE LP SOLUTION ARE FIRST.  JPERM PUTS THE NONBASIC   *****
1201.      C      *****   VARIABLES FIRST. JPERM(1),...,JPERM(N1) ARE THE GROUP 1  *****
1202.      C      *****   VARIABLES.                                               *****
1203.      C
1204.            N1=N-NB
1205.            IF(NB.EQ.N) GO TO 14
1206.            DO 221 J=1,N1
1207.              JPERM(J)=NB+J
1208.      221     CONTINUE
1209.      14    CONTINUE
1210.            LL=N1+1
1211.            DO 222 J=LL,N
1212.              JPERM(J)=J-N1
1213.      222     CONTINUE
1214.      C
1215.      C      *****   REVISE JPERM SO JPERM(K) IS THE KTH         *****
1216.      C      *****   SMALLEST U(JPERM(K)) FOR K=1,...,N1.  NOTE *****
1217.      C      *****   THAT U(-) IS THE SIZE OF THE RANGE OF       *****
1218.      C      *****   THE VALUES FOR THESE VARIABLES.             *****
1219.      C
1220.            IF(NB.EQ.N)GO TO 15
1221.            DO 223 K=1,N1
1222.              KK=JPERM(K)
1223.              MIN=U(KK)
1224.              DO 224 I=K,N1
1225.                JJ=JPERM(I)
1226.                IF(MIN.LE.U(JJ))GO TO 224
1227.                MIN=U(JJ)
1228.                JPERM(I)=JPERM(K)
1229.                JPERM(K)=JJ
1230.      224       CONTINUE
1231.      223     CONTINUE
1232.      15    CONTINUE
1233.            LL=N1+1
1234.      C
1235.      C      *****   STORE THE SIZE OF THE RANGE OF VALUES FOR X(J) IN TEMP(J).   *****
1236.      C
1237.            DO 225 J=1,N
1238.              TEMP(J)=U(J)-AL(J)
1239.      225     CONTINUE
1240.      C
1241.      C      *****   REVISE JPERM SO JPERM(N1+K) IS THE KTH      *****
1242.      C      *****   SMALLEST TEMP(JPERM(K)) FOR K=N1,...,N.     *****
1243.      C
1244.            DO 226 K=LL,N
1245.              KK=JPERM(K)
1246.              MIN=TEMP(KK)
1247.              DO 227 I=K,N
1248.                JJ=JPERM(I)
1249.                IF(MIN.LE.TEMP(JJ)) GO TO 227
1250.                MIN=TEMP(JJ)
1251.                JPERM(I)=JPERM(K)
1252.                JPERM(K)=JJ
1253.      227       CONTINUE
1254.      226     CONTINUE
1255.      C
1256.      C      *****   LPERM IS THE INVERSE PERMUTATION OF JPERM.   *****
1257.      C      *****   LPERM(J)=I MEANS VARIABLE J IN THE GROUP     *****
1258.      C      *****   ORDERING IS IN POSITION I IN THE JB ORDERING.*****
1259.      C
```

```
1260.              DO 231 J=1,N
1261.                LL=JPERM(J)
1262.                LPERM(LL)=J
1263.        231     CONTINUE
1264.        C
1265.        C       *****   REORDER X ACCORDING TO JPERM.   *****
1266.        C
1267.              DO 232 J=1,N
1268.                TEMP(J)=X(J)
1269.        232     CONTINUE
1270.              DO 233 J=1,N
1271.                LL=JPERM(J)
1272.                X(J)=TEMP(LL)
1273.        233     CONTINUE
1274.              LK=N+1
1275.        C
1276.        C       *****   REORDER THE COLUMNS OF EACH EXTREME     *****
1277.        C       *****   POINT ACCORDING TO JPERM.               *****
1278.        C
1279.              DO 234 I=1,LK
1280.                DO 235 J=1,N
1281.                  TEMP(J)=XX(I,J)
1282.        235       CONTINUE
1283.                DO 236 J=1,N
1284.                  LL=JPERM(J)
1285.                  XX(I,J)=TEMP(LL)
1286.        236       CONTINUE
1287.        234     CONTINUE
1288.        C
1289.        C       *****   REORDER THE LOWER BOUNDS ACCORDING TO JPERM.   *****
1290.        C
1291.              DO 241 J=1,N
1292.                TEMP(J)=AL(J)
1293.        241     CONTINUE
1294.              DO 242 J=1,N
1295.                LL=JPERM(J)
1296.                AL(J)=TEMP(LL)
1297.        242     CONTINUE
1298.        C
1299.        C       *****   REORDER THE UPPER BOUNDS ACCORDING TO JPERM.   *****
1300.        C
1301.              DO 243 J=1,N
1302.                TEMP(J)=U(J)
1303.        243     CONTINUE
1304.              DO 244 J=1,N
1305.                LL=JPERM(J)
1306.                U(J)=TEMP(LL)
1307.        244     CONTINUE
1308.        C
1309.        C       *****   FIX THE VARIABLES WITH 0 RANGES AT THEIR    *****
1310.        C       *****   UPPER BOUND(=LOWER BOUND).  PUT THE FIRST   *****
1311.        C       *****   TWO BASIC VARIABLES WITH POSITIVE RANGES IN *****
1312.        C       *****   GROUP 2.  LET N2 EQUAL THE INDEX OF THE     *****
1313.        C       *****   LAST GROUP 2 VARIABLE.                      *****
1314.        C
1315.              LL=N1+1
1316.              DO 245 J=LL,N
1317.                IF(U(J)-AL(J).LE.0.0) GO TO 245
1318.                K=J-1
1319.                GO TO 246
```

```
      245     CONTINUE
              K=N
      246     N2=K+2
              IF(K.EQ.N1)GO TO 247
              DO 248 J=LL,K
                X(J)=U(J)
      248       CONTINUE
      247     CONTINUE
C
C     *****    DIVIDE THE REMAINING VARIABLES EQUALLY        *****
C     *****    BETWEEN GROUPS 3 AND 4, PUTTING THOSE WITH    *****
C     *****    THE SMALLEST RANGES IN GROUP 3.  LET N3       *****
C     *****    EQUAL THE INDEX OF THE LAST GROUP 3 VARIABLE.*****
C
              IF(NB.EQ.3.OR.NB.EQ.4) N2=N1
              IF(NB.EQ.2)N2=N
              T=(N-N2)/2
              IF(T.NE.AINT(T)) GO TO 249
              N3=(N+N2)/2
              GO TO 251
      249     CONTINUE
              N3=(N+N2-1)/2
      251     CONTINUE
              RETURN
              END
C
C
C
C
C
              SUBROUTINE PART5
C
C
C     ****************************************************************
C     *      OBTAIN NEW LOWER AND UPPER BOUNDS FOR VARIABLES KL THROUGH NN      *
C     *      BASED ON THE FIXED VALUES OF X(1),...,X(N1) AND THE CURRENT BEST   *
C     *      FEASIBLE SOLUTION ZB.                                              *
C     ****************************************************************
C
C
              DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
             1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
             2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
             3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
             4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
             5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
              COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
             XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
             XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
             XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
             XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
             XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
C
C     *****    STEP B(REFERENCE 2, P. 28).    *****
C
              LA=N1+1
              LL=NB+1
              DO 80 I=1,N1
                KK=NB+I
                TEMP(I)=RHO(KK)
```

```
      80      CONTINUE
              W=RHP
              DO 310 I=LL,N
                K=LPERM(I)
      310       W=W+RHO(K)
              T=1.0-W
              C
              C     *****   FIND NEW BOUNDS.  LZ=1 YIELDS UPPER BOUNDS UEP(J);   *****
              C     *****   LZ=2 YIELDS LOWER BOUNDS ALEP(J).                    *****
              C
              DO 254 J=KL,NN
                DO 21 LZ=1,2
                  II=-1
                  DO 253 K=LA,N
                    JPERM(K)=K-N1
      253         CONTINUE
                  SUM=0.0
                  DO 255 K=LA,N
                    KK=JPERM(K)
                    MAX=XX(KK,J)
                    DO 256 I=K,N
                      JJ=JPERM(I)
                      IF(LZ.EQ.2)GO TO 30
                      IF(MAX.GE.XX(JJ,J))GO TO 256
                      GO TO 31
      30              IF(MAX.LE.XX(JJ,J))GO TO 256
      31              MAX=XX(JJ,J)
                      JPERM(I)=JPERM(K)
                      JPERM(K)=JJ
      256           CONTINUE
                    JJ=N+1
                    KK=JPERM(K)
                    TEMP(K)=Y(KK)
                    IF(LZ.EQ.2)GO TO 32
                    IF(XX(JJ,J).GE.XX(KK,J))GO TO 257
                    GO TO 33
      32            IF(XX(JJ,J).LE.XX(KK,J))GO TO 257
      33            SUM=SUM+Y(KK)
                    GO TO 255
      257           II=K
                    GO TO 258
      255           CONTINUE
              C
              C     *****   AT THIS POINT WE KNOW THAT PERMUTATION UP TO         *****
              C     *****   ELEMENT II IS POSSIBLY ENOUGH.  II=N1+N'+1 SO WE     *****
              C     *****   CAN FIND THE NEW WEIGHT ON XX(N+1,-) AND STORE       *****
              C     *****   IT IN SUM.                                           *****
              C
      258       R=T-SUM
                IF(R.LT.0.0)R=0.0
                SUM=RHP+R
                JJ=N+1
                THETA=SUM*XX(JJ,J)
                IF(N.EQ.NB)GO TO 41
                DO 259 K=1,N1
                  KK=NB+K
                  SUM=SUM+TEMP(K)
                  IF(SUM-1.0)261,262,263
      261         THETA=THETA+TEMP(K)*XX(KK,J)
                  GO TO 259
```

```
      262          THETA=THETA+TEMP(K)*XX(KK,J)
                   GO TO 264
      263          SUM=SUM-TEMP(K)
                   THETA=THETA+(1.0-SUM)*XX(KK,J)
                   GO TO 264
      259          CONTINUE
      41         CONTINUE
                 KK=N1+1
                 IF(II.EQ.-1)II=N
                 IF(KK.GT.II)GO TO 301
                 DO 265 K=KK,II
                   LL=JPERM(K)
                   SUM=SUM+TEMP(K)
                   IF(SUM-1.0)266,267,268
      266          THETA=THETA+TEMP(K)*XX(LL,J)
                   GO TO 265
      267          THETA=THETA+TEMP(K)*XX(LL,J)
                   GO TO 264
      268          SUM=SUM-TEMP(K)
                   THETA=THETA+(1.0-SUM)*XX(LL,J)
                   GO TO 264
      265          CONTINUE
      301        CONTINUE
C
C     *****    IF NECESSARY CONTINUE THE PERMUTA-    *****
C     *****    TION PAST II TO FIND THETA.            *****
C
                 II=II+1
                 IF(II.GT.N) GO TO 264
                 DO 269 K=II,N
                   KK=JPERM(K)
                   MAX=XX(KK,J)
                   DO 271 I=K,N
                     JJ=JPERM(I)
                     IF(LZ.EQ.2)GO TO 35
                     IF(MAX.GE.XX(JJ,J))GO TO 271
                     GO TO 36
      35             IF(MAX.LE.XX(JJ,J))GO TO 271
      36             MAX=XX(JJ,J)
                     JPERM(I)=JPERM(K)
                     JPERM(K)=JJ
      271            CONTINUE
                   LL=JPERM(K)
                   SUM=SUM+Y(LL)
                   IF(SUM-1.0)272,273,274
      272          THETA=THETA+Y(LL)*XX(LL,J)
                   GO TO 269
      273          THETA=THETA+Y(LL)*XX(LL,J)
                   GO TO 264
      274          SUM=SUM-Y(LL)
                   THETA=THETA+(1.0-SUM)*XX(LL,J)
                   GO TO 264
      269          CONTINUE
C
C     *****    SET THE NEW BOUNDS BASED ON THETA.    *****
C
      264        IF(LZ.EQ.2)GO TO 22
                 UEP(J)=AINT(THETA+.01)
                 GO TO 21
      22         IF(THETA.GT.0.0)GO TO 23
```

```
                THETA=0.0
                GO TO 24
      23        IF(THETA.NE.AINT(THETA))THETA=AINT(THETA)+1.0
      24        ALEP(J)=THETA
      21        CONTINUE
     254      CONTINUE
              RETURN
              END
C
C
C
C
C
      SUBROUTINE PART6
C
C
C     ****************************************************************************
C     *      INITIALIZE THE PROCEDURE FOR THE GROUP 1 VARIABLES.                 *
C     ****************************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(51),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,LBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
C
C     *****   NZERP IS LARGEST J SUCH THAT U(J)=0, J=1,...,N1.    *****
C
      IF(N1.EQ.0)GO TO 10
      IF(U(N1))301,301,303
  301 NZERP=N1
      GO TO 307
  303 DO 302 J=1,N1
        IF(U(J).LE.0.0)GO TO 302
        NZERP=J-1
        GO TO 307
  302   CONTINUE
C
C     *****   SET RHO(J)=0.0, J=1,...,N1.  RHO(J) IS THE          *****
C     *****   CURRENT WEIGHT ON EXTREME POINT J.  FIX THIS        *****
C     *****   VALUE FOR J=1,...,NZERP.  SET X(I)=0.0, I=1,...N1   *****
C     *****   AS THE INITIAL ELIGIBLE PARTIAL SOLUTION FOR THE    *****
C     *****   GROUP 1 VARIABLES.                                  *****
C
  307 DO 304 J=1,N1
        X(J)=0.0
        RHO(J)=0.0
  304   CONTINUE
      INDIC=0
      IF(NZERP.LT.N1)RETURN
C
C     *****   INDIC=1 MEANS THIS IS THE ONLY ELIGIBLE PARTIAL     *****
```

```
1560.      C      *****   SOLUTION FOR THE GROUP 1 VARIABLES.                *****
1561.      C
1562.      10     INDIC=1
1563.             RETURN
1564.             END
1565.      C
1566.      C
1567.      C
1568.      C
1569.      C
1570.             SUBROUTINE PART7(LARGE)
1571.      C
1572.      C
1573.      C      ************************************************************************
1574.      C      *      OBTAIN A NEW ELIGIBLE PARTIAL SOLUTION FOR THE GROUP 1 VARIABLES.  *
1575.      C      *      VARIABLES X(1),...,X(LARGE) HAVE ALREADY BEEN SCANNED.             *
1576.      C      ************************************************************************
1577.      C
1578.      C
1579.             DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
1580.            1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
1581.            2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
1582.            3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
1583.            4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
1584.            5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
1585.             COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
1586.            XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
1587.            XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
1588.            XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
1589.            XULP,UMIN,UU,JT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
1590.            XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
1591.      201    FORMAT('-','THE ELIGIBLE GROUP 1 SOLUTIONS HAVE BEEN SCANNED',
1592.            X' THROUGH VARIABLE',I4,'.')
1593.      200    FORMAT(1H ,15I5)
1594.             IF(INDIC.EQ.1.OR.N1.EQ.0) GO TO 315
1595.             RHP=(ZB-ZF)/(ZX-ZF)
1596.             T=RHP-1.0
1597.             LL=NZERP+1
1598.      C
1599.      C      *****   RECALL JPERM ORDERS THE VARIABLES INTO          *****
1600.      C      *****   GROUPS, RHO IS INDEXED ACCORDING TO THE JB      *****
1601.      C      *****   ORDERING SINCE THE EXTREME POINTS ARE IN        *****
1602.      C      *****   THAT ORDER.                                     *****
1603.      C
1604.             DO 311 J=LL,N1
1605.               K=JPERM(J)
1606.               T=T+RHO(K)
1607.      311      CONTINUE
1608.             DO 316 J=LL,N1
1609.               IF(X(J).LT.U(J).OR.J.LE.LARGE)GO TO 400
1610.               LARGE=J
1611.               WRITE(6,201) LARGE
1612.      400      CONTINUE
1613.               IF(X(J)-U(J))312,313,313
1614.      C
1615.      C      *****   X(J) CAN BE INCREASED, DO SO AND ADJUST RHO(JPERM(J)).   *****
1616.      C
1617.      312      X(J)=X(J)+1.0
1618.               K=JPERM(J)
1619.               IF(XX(K,J).EQ.0.0) WRITE(6,200) K,J
```

```
1620.                TT=1.0/XX(K,J)
1621.                RHO(K)=RHO(K)+TT
1622.                T=T+TT
1623.                IF(T)314,314,313
1624.      313       IF(J.EQ.N1)GO TO 315
1625.      C
1626.      C     *****   AN ELIGIBLE SOLUTION REQUIRED GOING TO J+1 AND   *****
1627.      C     *****   SETTING X(J)=RHO(JPERM(J))=0.0.                  *****
1628.      C
1629.                X(J)=0.0
1630.                K=JPERM(J)
1631.                T=T-RHO(K)
1632.                RHO(K)=0.0
1633.      316       CONTINUE
1634.      C
1635.      C     *****   STEP 314 MEANS AN ELIGIBLE SOLUTION HAS BEEN FOUND.  *****
1636.      C     *****   STEP 315 MEANS NO ELIGIBLE SOLUTION EXISTS.          *****
1637.      C
1638.      314   CONTINUE
1639.            NOSOLN=1
1640.            RETURN
1641.      315   CONTINUE
1642.            NOSOLN=0
1643.            RETURN
1644.            END
1645.      C
1646.      C
1647.      C
1648.      C
1649.      C
1650.            SUBROUTINE PART8
1651.      C
1652.      C
1653.      C     ***************************************************************************
1654.      C     *      IF ONE EXISTS OBTAIN AN INITIAL TRIAL PARTIAL SOLUTION FOR THE     *
1655.      C     *      GROUP 2 VARIABLES, GIVEN X(1),...,X(N1) AND ZB.  ALSO INITIALIZE   *
1656.      C     *      THE PROCEDURE FOR THE GROUP 2 VARIABLES.                           *
1657.      C     ***************************************************************************
1658.      C
1659.      C
1660.            DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
1661.           1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
1662.           2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
1663.           3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
1664.           4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
1665.           5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
1666.            COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
1667.           XINDEXC,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
1668.           XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
1669.           XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
1670.           XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
1671.           XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
1672.            LL=N1+1
1673.            DO 321 J=LL,N2
1674.              IF(UEP(J).LT.ALEP(J)) GO TO 322
1675.      321     CONTINUE
1676.            GO TO 323
1677.      C
1678.      C     *****   ARRIVAL AT LABEL 322 INDICATES THAT *****
1679.      C     *****   NO ELIGIBLE PARTIAL SOLUTION EXISTS.*****
```

```
      C
  322 CONTINUE
      SOLN=0.0
      RETURN
  323 LL=N2-1
      C
      C     *****   STEP 8(REFERENCE 2, P. 34).  AN INITIAL          *****
      C     *****   PARTIAL SOLUTION EXISTS, FIND IT.  DELTA(J)      *****
      C     *****   =1.0 MEANS THAT X(J) IS AT ITS UPPER BOUND.      *****
      C
      DO 324 J=LL,N2
        IF(XB(J).LE.UEP(J))GO TO 325
        X(J)=UEP(J)
        DELTA(J)=1.0
        GO TO 324
  325   IF(XB(J).GE.ALEP(J))GO TO 326
        X(J)=ALEP(J)
        DELTA(J)=0.0
        GO TO 324
  326   X(J)=XB(J)
        DELTA(J)=0.0
  324   CONTINUE
      C
      C     *****   STORE THE INITIAL GROUP 2 PARTIAL          *****
      C     *****   SOLUTION DATA FOR FUTURE USE.              *****
      C
      XPN2M1=X(LL)
      XPN2=X(N2)
      DN2P=DELTA(N2)
      C
      C     *****   FIND QUANTITIES TO BE USED IN SUBROUTINE PART10.  *****
      C     *****                 RN2M1 = R(N2-1)                     *****
      C     *****                 RN2 = R(N2)                         *****
      C     *****                 UU(NB+1) = U(0)                     *****
      C     *****                 VV(NB+1) = V(0)                     *****
      C     *****                 UU(I) = U(I)                        *****
      C     *****                 VV(I) = V(I)                        *****
      C     *****   QUANTITIES ON THE RIGHT ARE FROM REFERENCE 2.      *****
      C     *****   P. 35.                                            *****
      C
      LL=NB+1
      KK=N+1
      JJ=N2-1
      RN2M1=RHP*XX(KK,JJ)
      RN2=RHP*XX(KK,N2)
      W=RHP
      IF(NB.EQ.N)GO TO 40
      DO 252 I=LL,N
        RN2M1=RN2M1+XX(I,JJ)*RHO(I)
        RN2=RN2+XX(I,N2)*RHO(I)
        W=W+RHO(I)
  252   CONTINUE
   40 CONTINUE
      T=1.0-W
      KK=NB+1
      LL=N+1
      JJ=N2-1
      UU(KK)=T*XX(LL,JJ)+RN2M1
      UBAR=U(KK)
      VV(KK)=T*XX(LL,N2)+RN2
```

```
      VBAR=VV(KK)
      DO 327 I=1,N9
        UU(I)=T*XX(I,JJ)+RN2M1
        UBAR=UBAR+UU(I)
        VV(I)=T*XX(I,N2)+RN2
        VBAR=VBAR+VV(I)
327   CONTINUE
      SOLN=1.0
      RETURN
      END
C
C
C
C
      SUBROUTINE PART9
C
C
C     ****************************************************************************
C     *      IF ONE EXISTS OBTAIN A NEW TRIAL PARTIAL SOLUTION FOR THE GROUP 2   *
C     *      VARIABLES, GIVEN X(1),...,X(N1) AND ZB.                             *
C     ****************************************************************************
C
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(51),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,N9,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SQRT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      J=N2
331   IF(DELTA(J).EQ.1.0) GO TO 332
      IF(X(J).EQ.UEP(J)) GO TO 333
C
C     *****   STEP C(REFERENCE 2, PP. 35-36).   *****
C
      X(J)=X(J)+1.0
      GO TO 334
333   DELTA(J)=1.0
      IF(J.EQ.N2) GO TO 335
      X(J)=XPN2M1
      GO TO 332
335   X(J)=XPN2
C
C     *****   STEP D(REFERENCE 2, P. 36).   *****
C
332   IF(X(J).EQ.ALEP(J))GO TO 336
      X(J)=X(J)-1.0
      GO TO 334
336   IF(J.NE.N2)GO TO 337
      X(J)=XPN2
      DELTA(J)=DN2P
```

```
1800.       C
1801.       C      *****   LABEL 334 INDICATES THAT A NEW TRIAL     *****
1802.       C      *****   PARTIAL SOLUTION EXISTS.                 *****
1803.       C
1804.       334    CONTINUE
1805.              SOLN=1.0
1806.              RETURN
1807.       C
1808.       C      *****   LABEL 337 INDICATES THAT NO NEW TRIAL      *****
1809.       C      *****   PARTIAL SOLUTION EXISTS.                   *****
1810.       C
1811.       337    CONTINUE
1812.              SOLN=0.0
1813.              RETURN
1814.              END
1815.       C
1816.       C
1817.       C
1818.       C
1819.       C
1820.              SUBROUTINE PART10
1821.       C
1822.       C
1823.       C      ************************************************************************
1824.       C      *      DETERMINE IF A GIVEN TRIAL PARTIAL SOLUTION FOR THE GROUP 2     *
1825.       C      *      VARIABLES IS ELIGIBLE.                                          *
1826.       C      ************************************************************************
1827.       C
1828.       C
1829.              DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
1830.             1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
1831.             2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
1832.             3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
1833.             4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
1834.             6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
1835.              COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
1836.             XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
1837.             XLPERM,LZ,M,MAX,MIN,N,N8,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
1838.             XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
1839.             XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
1840.             XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
1841.              JJ=N8+1
1842.              LL=N2-1
1843.       C
1844.       C      *****   STEP A(REFERENCE 2, P. 36).   *****
1845.       C
1846.              UT=X(LL)
1847.              VT=X(N2)
1848.       C
1849.       C      *****   STEP B(REFERENCE 2, PP. 36-37).   *****
1850.       C
1851.              IF(UT.EQ.UBAR.AND.VT.EQ.VBAR)GO TO 372
1852.              IF(UT.EQ.UBAR)GO TO 342
1853.              IF(VT.EQ.VBAR)GO TO 343
1854.       C
1855.       C      *****   STEP C(REFERENCE 2, P. 37).  NOTE THAT AAA = A     *****
1856.       C      *****   IN REFERENCE 2.                                    *****
1857.       C
1858.              SZERO=(VT-VBAR)/(UT-UBAR)
1859.              SONE=-1.0/SZERO
```

```
      AAA=SONE-SZERO
C
C     *****   STEP D(REFERENCE 2, P. 37).  NOTE    *****
C     *****   THAT T = TAN(THETA) IN REFERENCE 2.  *****
C
      IF(UT-UBAR-1.0.NE.0.0)GO TO 344
      T=-SZERO
      GO TO 345
  344 T=(VT-VBAR-SONE)/(UT-UBAR-1.0)
      T=(T-SONE)/(1.0+SONE*T)
  345 CONTINUE
C
C     *****   STEP F(REFERENCE 2, P. 38).  INDEXQ(I)=1     *****
C     *****   MEANS I IS AN ELEMENT OF L;  INDEXQ(I)=2     *****
C     *****   MEANS I IS AN ELEMENT OF R.  L AND R ARE     *****
C     *****   DEFINED IN REFERENCE 1, P. 655.              *****
C
      IF(T)346,346,347
  346 LL=1
      KK=2
      GO TO 348
  347 LL=2
      KK=1
  348 DO 353 I=1,JJ
        T=AAA*(VV(I)-VBAR-SZERO*(UU(I)-UBAR))
        IF(T)349,351,352
  349   INDEXQ(I)=LL
        GO TO 353
  351   INDEXQ(I)=0
        GO TO 353
  352   INDEXQ(I)=KK
  353   CONTINUE
      GO TO 354
  342 T=VT-VBAR
C
C     *****   STEP G(REFERENCE 2, P. 38).  INDEXQ(I)=1 MEANS  *****
C     *****   I IS AN ELEMENT OF L;  INDEXQ(I)=2 MEANS I IS   *****
C     *****   AN ELEMENT OF R.  L AND R ARE DEFINED IN REFER- *****
C     *****   ENCE 1, P. 655.                                 *****
C
      IF(T)355,355,356
  355 LL=1
      KK=2
      GO TO 357
  356 LL=1
      KK=1
  357 DO 358 I=1,JJ
        T=UT-UU(I)
        IF(T)359,361,362
  359   INDEXQ(I)=LL
        GO TO 358
  361   INDEXQ(I)=0
        GO TO 358
  362   INDEXQ(I)=KK
  358   CONTINUE
      GO TO 354
C
C     *****   STEP H(REFERENCE 2, P. 38).  INDEXQ(I)=1 MEANS  *****
C     *****   I IS AN ELEMENT OF L;  INDEXQ(I)=2 MEANS I IS   *****
C     *****   AN ELEMENT OF R.  L AND R ARE DEFINED IN REFER- *****
```

```
C      *****   ENCE 1, P. 655.                                          *****
C
343    T=UT-UBAR
       IF(T)363,363,364
363    LL=2
       KK=1
       GO TO 365
364    LL=1
       KK=2
365    DO 366 I=1,JJ
         T=VT-VV(I)
         IF(T)367,368,369
367      INDEXQ(I)=LL
         GO TO 366
368      INDEXQ(I)=0
         GO TO 366
369      INDEXQ(I)=KK
366      CONTINUE
C
C      *****   STEP I(REFERENCE 2, PP. 38-39).   *****
C
354    T=UT-UBAR
       TT=VT-VBAR
       DO 371 I=1,JJ
         IF(INDEXQ(I).NE.0) GO TO 371
         IF(ABS(UU(I)-UBAR).LT.ABS(T).OR.T*(UU(I)-UBAR).LT.0.0)GO TO 371
         IF(ABS(VV(I)-VBAR).LT.ABS(TT).OR.TT*(VV(I)-VBAR).LT.0.0) GO
     X   TO 371
         GO TO 372
371      CONTINUE
       GO TO 373
C
C      *****   ARRIVAL AT LOCATION 372 MEANS THE      *****
C      *****   TRIAL PARTIAL SOLUTION IS ELIGIBLE.    *****
C
372    CONTINUE
       SOLN=1.0
       RETURN
C
C      *****   STEPS J,K AND L(REFERENCE 2, P. 39).       *****
C      *****   STORE MAX(R) IN AMAX, MIN(L) IN AMIN.      *****
C
373    AMAX=-10000000.0
       AMIN=10000000.0
       DO 374 I=1,JJ
         IF(INDEXQ(I).EQ.0)GO TO 374
         IF(VT.NE.VBAR)GO TO 30
         IF(UT.NE.UU(I))GO TO 10
         T=0.0
         GO TO 379
10       IF(VT.EQ.VV(I))GO TO 11
         T=(UU(I)-UT)/(VT-VV(I))
         GO TO 379
11       T=SIGN(10.0**34,UU(I)-UT)
         GO TO 379
30       IF(UT.NE.UU(I))GO TO 15
         IF(UT.NE.UBAR)GO TO 16
         T=SIGN(10.0**34,VBAR-VT)
         GO TO 379
```

```
1980.                GO TO 379
1981.      15        SONE=(UBAR-UT)/(VT-VBAR)
1982.                T=(VT-VV(I))/(UT-UU(I))
1983.                T=(T-SONE)/(1.0+SONE*T)
1984.      379       IF(INDEXQ(I).EQ.1)GO TO 383
1985.                IF(T.GT.AMAX)AMAX=T
1986.                GO TO 384
1987.      383       IF(T.LT.AMIN)AMIN=T
1988.      384       IF(AMAX-AMIN)374,385,385
1989.      374       CONTINUE
1990.      C
1991.      C         *****    ARRIVAL AT 386 MEANS THE TRIAL          *****
1992.      C         *****    SOLUTION IS NOT ELIGIBLE.               *****
1993.      C
1994.      386       CONTINUE
1995.                SOLN=0.0
1996.                RETURN
1997.      C
1998.      C         *****    ARRIVAL AT 385 MEANS THE TRIAL          *****
1999.      C         *****    SOLUTION IS ELIGIBLE.                   *****
2000.      C
2001.      385       CONTINUE
2002.                GO TO 372
2003.                END
2004.      C
2005.      C
2006.      C
2007.      C
2008.      C
2009.                SUBROUTINE PART11
2010.      C
2011.      C
2012.      C         ****************************************************************
2013.      C         *      INITIALIZE THE BOUNDING PROCEDURE FOR GROUP 3 AND 4 VARIABLES.     *
2014.      C         ****************************************************************
2015.      C
2016.      C
2017.                DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
2018.               1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
2019.               2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
2020.               3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
2021.               4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
2022.               5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
2023.                COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
2024.               XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
2025.               XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,P,SUM,SUM1,SZERO,U,UEP,
2026.               XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
2027.               XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
2028.               XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,K N,D1,D2,Y1,Y2,THOLD
2029.                DIMENSION ABAR(61,181),BBAR(61),CBAR(181),XBAR(181),DUAL1(181),
2030.               1DUAL2(181),CD(181),JH(121)
2031.      C
2032.      C         *****    STEPS A AND B(REFERENCE 2, PP. 42-43).  STORE    *****
2033.      C         *****    -P(SUPERSCRIPT F) SUBSCRIPT I (AS DEFINED IN     *****
2034.      C         *****    REFERENCE 1, P. 659) IN CBAR(I).                 *****
2035.      C
2036.                LL=N2+1
2037.                LM=N3+1
2038.                J=N2
2039.      25        J=J+1
```

```
      IF(J.GT.LM)GO TO 40
C
C*****   THIS IS THE CASE WHERE J-1 = MIN(J-1,N3).    *****
C*****   STORE THE MIN IN NJ.                         *****
C
      LAST=N-J+1
      NJ=J-1
      T=-ZF
      IF(NJ.LE.0)GO TO 50
      DO 11 K=1,NJ
        LK=JPERM(K)
   11   T=T+C(LK)*XF(LK)
   50 CONTINUE
      THOLD(M+1)=-T
      CBAR(M+1)=-T
      CD(M+1)=T
      DO 12 I=1,M
        T=0.0
        IF(NJ.LE.0)GO TO 51
        DO 13 K=1,NJ
          LK=JPERM(K)
   13     T=T+A(I,LK)*XF(LK)
   51   CONTINUE
        THOLD(I)=T
        CBAR(I)=T-B(I)
        CD(I)=-CBAR(I)
   12   CONTINUE
      DO 14 I=1,LAST
        BBAR(I)=0.0
        LK=JPERM(N-I+1)
        ABAR(I,M+1)=C(LK)
        DO 15 K=1,M
   15     ABAR(I,K)=-A(K,LK)
   14   CONTINUE
      BBAR(LAST)=-1.0
C
C*****   PD=1.0 LEAD TO SOLUTION VIA THE SIMPLEX METHOD.   *****
C
      PD=1.0
      L1=N-J+1
      L2=M+1
C
C*****   SOLVE THE DUAL OF THE PROBLEM IN REFERENCE 1, P. 659.   *****
C
      CALL SOLVLP(ABAR,BBAR,CBAR,LAST,FJBF,GJBF,DUAL1,DUAL2,PD,L1,L2,
     XCD,JH)
   10 CONTINUE
   20 CONTINUE
C
C*****   Y1(J)   = Y*(J,0)                          *****
C*****   Y2(J)   = Y**(J,0)                         *****
C*****   DUAL1(I) = Y*(J,I)                         *****
C*****   DUAL2(I) = Y**(J,I)                        *****
C*****   D1(J,N+1) = D*(J,0)                        *****
C*****   D2(J,N+1) = D**(J,0)                       *****
C*****   D1(J,K) = D*(J,K)                          *****
C*****   D2(J,K) = D**(J,K)                         *****
C*****   FJBF = MAX X(J) S.T. CONSTRAINTS           *****
C*****   GJBF = MIN X(J) S.T. CONSTRAINTS           *****
C*****   THE VARIABLES ON THE RIGHT ARE             *****
```

```
C      *****   DEFINED IN REFERENCE 1, PP. 659-660*****
C      *****   AND REFERENCE 2, P. 43.              *****
C
      LR=M+1
      Y1(J)=DUAL1(M+1)
      Y2(J)=DUAL2(M+1)
      T=FJBF+Y1(J)*THOLD(M+1)
      DO 21 I=1,M
   21   T=T+DUAL1(I)*THOLD(I)
      D1(J,N+1)=T
      T=GJBF-Y2(J)*THOLD(M+1)
      DO 22 I=1,M
   22   T=T-DUAL2(I)*THOLD(I)
      D2(J,N+1)=T
      IF(NJ.LE.0)GO TO 52
      DO 23 K=1,NJ
        LK=JPERM(K)
        T=Y1(J)*C(LK)
        TT=-Y2(J)*C(LK)
        DO 24 I=1,M
          T=T-DUAL1(I)*A(I,LK)
          TT=TT+DUAL2(I)*A(I,LK)
   24     CONTINUE
        D1(J,K)=T
        D2(J,K)=TT
   23   CONTINUE
   52 CONTINUE
      LT=J
      IF(PD.EQ.1.0)GO TO 25
      GO TO 42
C
C      *****   THIS IS THE CASE WHERE N3 = MIN (J-1,N3).    *****
C      *****   STORE THE MIN IN NJ.  PD=0.0 LEADS TO SOLU-  *****
C      *****   TION VIA THE DUAL SIMPLEX METHOD.  THIS IS   *****
C      *****   POSSIBLE BECAUSE IN THIS CASE ABAR,BBAR AND  *****
C      *****   CBAR DON'T CHANGE.                           *****
C
   40 PD=0.0
      NJ=N3
      L1=N-N2
      L1=N-N3
      L2=M+1
   41 LAST=N-J+1
      CALL SOLVLP(ABAR,BBAR,CBAR,LAST,FJBF,GJBF,DUAL1,DUAL2,PD,L1,L2,
     1CD,JH)
      GO TO 20
   42 J=J+1
      IF(J.GT.N)RETURN
      GO TO 41
      END
C
C
C
C
      SUBROUTINE SOLVLP(ABAR,BBAR,CBAR,LAST,FJBF,GJBF,DUAL1,DUAL2,
     1PD,M,N,CD,JH)
C
C
C      ***********************************************************
```

```
C     ****************************************************************
C     **      THIS SUBROUTINE HAS THE CAPACITY TO SOLVE LINEAR PROGRAMS      **
C     **      USING THE SIMPLEX OR DUAL SIMPLEX METHOD.  THE METHODS ARE     **
C     **      DESCRIBED IN REFERENCE 4.  IT IS CALLED FROM PART11 TO SOLVE FOR **
C     **      THE VARIABLES DESCRIBED THERE AND IN REFERENCE 1, PP. 659-660.  **
C     **      PD TELLS IT WHICH METHOD AND OBJECTIVE FUNCTION TO USE IN       **
C     **      SOLVING THE LP WITH CONSTRAINT MATRIX ABAR AND RIGHT HAND SIDE  **
C     **      BBAR.                                                          **
C     ****************************************************************
C     ****************************************************************
C
      DIMENSION ABAR(61,181),A(61,181),INFIX(8),TOL(4),BBAR(61),
     1CBAR(181),XBAR(181),BB(61,61),QSTAR(61),XL(121),X(121),CPERM(181),
     20(61),S(61),JH(121),KOUT(7),ERR(8),KB(181),P(181),Y(181),
     3DUAL1(181),DUAL2(181),NEGB(61),CO(181)
      LOGICAL SIGN
  501 FORMAT(19H UNBOUNDED VARIABLE)
      NPRIME=0
      K2=N
      NSAVE=N
      K3=M+N
      COUNT1=0.0
C
C     *****   PD=1.0 LEADS TO SOLUTION VIA THE SIMPLEX METHOD.   *****
C
      IF(PD.NE.1.0) GO TO 1500
C
C     *****   THE FOLLOWING SETS UP THE DATA FOR LP   *****
C     *****   ROUTINE.  FIRST, ADD SLACK VARIABLES.   *****
C
      DO 1406 I=1,M
        DO 1409 J=1,M
 1409     ABAR(I,N+J)=0.0
 1406   CONTINUE
      DO 1410 I=1,M
        ABAR(I,N+I)=1.0
 1410   CBAR(N+I)=0.0
C
C     *****   STORE THE COST COEFFICIENTS IN THE FIRST ROW OF ABAR.   *****
C
      DO 1412 I=1,M
        I1=M-I+1
        I2=M-I+2
        LML=I+1
        QSTAR(LML)=BBAR(I)
        DO 1413 J=1,K3
          ABAR(I2,J)=ABAR(I1,J)
 1413     CONTINUE
 1412   CONTINUE
      QSTAR(1)=0.0
      DO 1414 J=1,K3
 1414   ABAR(1,J)=-CBAR(J)
C
C     *****   ADD ARTIFICIAL VARIABLES WHEN BBAR(I) .LE. 0.   *****
C
      FINALN=N+M
      DO 9995 I=1,M
 9995   NEGB(I)=0
      LL=M+1
```

```
2220.              T=K3
2221.              DO 9990 I=2,LL
2222.                IF(QSTAR(I).GE.0.)GO TO 9990
2223.                T=T+1
2224.                DO 9991 J=1,K3
2225.      9991        ABAR(I,J)=-ABAR(I,J)
2226.                QQ=I-1
2227.                NEGB(QQ)=1
2228.                BBAR(QQ)=-BBAR(QQ)
2229.                QSTAR(I)=-QSTAR(I)
2230.                DO 9992 J=2,LL
2231.      9992        ABAR(J,T)=0.0
2232.      C
2233.      C       *****   USE THE BIG M METHOD ON THE ARTIFICIAL VARIABLES.    *****
2234.      C
2235.                ABAR(I,T)=1.0
2236.                ABAR(1,T)=1000
2237.      9990      CONTINUE
2238.      C
2239.      C       *****   FINALN CONTAINS THE NUMBER OF VARIABLES FOR LP ROUTINE.    *****
2240.      C
2241.              FINALN=T
2242.              INFIX(1)=4
2243.              INFIX(2)=FINALN
2244.              INFIX(3)=61
2245.              INFIX(4)=M+1
2246.              INFIX(5)=2
2247.              INFIX(6)=1
2248.              INFIX(7)=500
2249.              INFIX(8)=0
2250.              TOL(1)=10.0**(-6)
2251.              TOL(2)=10.0**(-5)
2252.              TOL(3)=-10.0**(-3)
2253.              TOL(4)=10.0**(-10)
2254.              PRM=0.0
2255.              COUNT1=COUNT1+1.0
2256.      C
2257.      C       *****   FIND THE LP SOLUTION.    *****
2258.      C
2259.              CALL SIMPLX(INFIX,ABAR,QSTAR,TOL,PRM,KOUT,ERR,JH,XL,P,Y,KB,BB)
2260.              IF(KOUT(1).EQ.3)GO TO 10
2261.              IF(KOUT(1).EQ.5)WRITE(6,501)
2262.              STOP
2263.      10      CONTINUE
2264.      C
2265.      C       *****   THE FOLLOWING RESTORES THE ABAR MATRIX    *****
2266.      C       *****   TO ORIGINAL FORM.                         *****
2267.      C
2268.              DO 1415 I=1,M
2269.                LML=I+1
2270.                DO 1416 J=1,K3
2271.                  ABAR(I,J)=ABAR(LML,J)
2272.      1416      CONTINUE
2273.      1415      CONTINUE
2274.      C
2275.      C       *****   THE FOLLOWING RESTORES THE ROWS OF THE INVERSE     *****
2276.      C       *****   AND ABAR NEGATED DUE TO NEGATIVE B(I).             *****
2277.      C
2278.              DO 9993 I=1,M
2279.                IF(NEGB(I).LE.0)GO TO 9993
```

```
2280.                DO 9994 J=1,N
2281.      9994        ABAR(I,J)=-ABAR(I,J)
2282.                DO 9996 K=2,LL
2283.                  II=I+1
2284.      9996        BB(II,K)=-BB(II,K)
2285.                BBAR(I)=-BBAR(I)
2286.      9993      CONTINUE
2287.              LL=M+1
2288.      C
2289.      C       *****   STORE THE LP SOLUTION IN XBAR.   *****
2290.      C
2291.              DO 1417 J=1,K3
2292.                LML=KB(J)
2293.                IF(LML.GT.0) GO TO 1418
2294.                XBAR(J)=0.0
2295.                GO TO 1417
2296.      1418      XBAR(J)=XL(LML)
2297.      1417      CONTINUE
2298.      C
2299.      C       *****   RETRIEVE THE OPTIMAL BASIS INVERSE.   *****
2300.      C
2301.              JJ=1
2302.              II=0
2303.              DO 1420 J=1,LL
2304.                DO 1421 I=1,LL
2305.                  II=II+1
2306.                  IF(II.LE.LL)GO TO 1422
2307.                  II=1
2308.                  JJ=JJ+1
2309.      1422        CONTINUE
2310.                  A(I,N+J)=BB(II,JJ)
2311.      1421        CONTINUE
2312.      1420      CONTINUE
2313.              DO 1423 I=1,M
2314.                DO 1424 J=1,M
2315.                  BB(I,J)=A(I+1,N+J+1)
2316.      1424        CONTINUE
2317.      1423      CONTINUE
2318.      C
2319.      C       *****   STORE THE LP SOLUTION IN DUAL1.  *****
2320.      C
2321.              DO 1425 J=1,K2
2322.      1425      X(NPRIMF+J)=XBAR(J)
2323.              DO 175 J=1,K2
2324.      175       DUAL1(J)=XBAR(J)
2325.              T=0.0
2326.              DO 8003 J=1,K2
2327.      8003      T=T+DUAL1(J)*CD(J)
2328.              FJBF=T
2329.      C
2330.      C       *****   FIND THE UPDATED SIMPLEX TABLEAU.   *****
2331.      C
2332.              DO 1426 I=1,M
2333.                II=JH(I+1)
2334.      C
2335.      C       *****   CPERM(I) IS THE COST COEFFICIENT FOR   *****
2336.      C       *****   THE ITH BASIC VARIABLE.               *****
2337.      C
2338.                CPERM(I)=CBAR(II)
2339.      1426      CONTINUE
```

```
      DO 1427 I=1,M
        T=0.0
        DO 1428 K=1,M
          T=T+CPERM(K)*BB(K,I)
1428    CONTINUE
C
C     *****   Q(I) IS THE MULTIPLIER FOR EQUATION I.   *****
C
        Q(I)=T
1427    CONTINUE
      DO 1429 J=1,K3
        IF(KB(J).GT.0) GO TO 1430
        T=0.0
        DO 1714 I=1,M
          T=T+Q(I)*ABAR(I,J)
1714    CONTINUE
        CBAR(J)=CBAR(J)-T
        GO TO 1429
1430    CBAR(J)=0.0
1429    CONTINUE
      DO 1457 J=1,K3
1457    CBAR(J)=-CBAR(J)
      DO 1434 J=1,K2
        DO 1435 I=1,M
          T=0.0
          DO 1436 K=1,M
1436        T=T+BB(I,K)*ABAR(K,J)
          S(I)=T
1435      CONTINUE
        DO 1437 I=1,M
1437      ABAR(I,J)=S(I)
1434    CONTINUE
      DO 1438 J=1,M
        LML=K2+J
        DO 1439 I=1,M
          ABAR(I,LML)=BB(I,J)
1439    CONTINUE
1438    CONTINUE
C
C     *****   THIS REVISES JH FOR USE IN THE DUAL SIMPLEX          *****
C     *****   METHOD (RECALL THAT IN THE SIMPLEX ROUTINE JH(I)     *****
C     *****   IS THE (I-1)ST BASIC VARIABLE).                      *****
C
      DO 1456 I=1,M
1456    JH(I)=JH(I+1)
      GO TO 1500
20    CONTINUE
C
C     *****   STORE THE SOLUTION IN DUAL1 AND COMPUTE THE      *****
C     *****   OBJECTIVE FUNCTION VALUE FJBF.                   *****
C
      DO 907 J=1,N
907     DUAL1(J)=XBAR(J)
      T=0.0
      DO 1512 J=1,N
1512    T=T+DUAL1(J)*CD(J)
      FJBF=T
C
C     *****   NOW PREPARE TO USE THE DUAL SIMPLEX METHOD.  WE   *****
C     *****   USE IT TO FIND GJBF WHEN PD=1.0 OR WHEN PD=0.0   *****
```

```
C      *****   AND COUNT=1.0, AND TO FIND FJBF WHEN PD=0.0 AND  *****
C      *****   COUNT=1.0.                                       *****
C
1500  CONTINUE
      DO 1448 I=1,M
1448    S(I)=0.0
      S(LAST)=1.0
      IF(COUNT1.EQ.0.0) S(LAST)=-1.0
      DO 1450 J=1,M
        LML=K2+J
        DO 1451 I=1,M
          BB(I,J)=ABAR(I,LML)
1451      CONTINUE
1450    CONTINUE
      DO 1452 I=1,M
        T=0.0
        DO 1453 J=1,M
1453      T=T+BB(I,J)*S(J)
        BBAR(I)=T
1452    CONTINUE
1404  CONTINUE
      COUNT1=COUNT1+1.0
      SIGN=.FALSE.
C
C      *****   FIND THE DUAL SIMPLEX SOLUTION.   *****
C
      CALL DUAL(ABAR,BBAR,CBAR,JH,XBAR,Y,M,K3,SIGN)
      IF(PD.EQ.1.0) GO TO 1400
      IF(COUNT1.EQ.1.0) GO TO 20
1400  CONTINUE
C
C      *****   STORE THE SOLUTION IN DUAL2 AND COMPUTE THE      *****
C      *****   OBJECTIVE FUNCTION VALUE GJBF.                   *****
C
      DO 903 J=1,N
903     DUAL2(J)=XBAR(J)
      T=0.0
      DO 1511 J=1,N
1511    T=T+DUAL2(J)*CD(J)
      GJBF=-T
      RETURN
      END
C
C
C
C
C
      SUBROUTINE SIMPLX(INFIX,A,B,TOL,PRM,KOUT,ERR,JH,X,P,Y,KB,E)
C
C
C      ************************************************************************
C      *      THE PURPOSE OF THIS CODE IS TO FIND THE SOLUTION OF THE LP PROBLEM:*
C      *                                                                        *
C      *                          MAXIMIZE     XZERO = CX                       *
C      *                  SUBJECT TO:                                           *
C      *                               AX . LE . B                              *
C      *                                X . GE . 0                              *
C      *                                                                        *
C      ************************************************************************
C
```

```
      C
       DIMENSION INFIX(8),A(61,181),B(61),TOL(4),KOUT(7),ERR(8),ZZ(3),
     1JH(121),X(121),P(181),Y(181),KB(181),E(61,61),TERR(8),IOFIX(16)
       EQUIVALENCE (INFLAG,IOFIX(1)),(N,IOFIX(2)),
     1(ME,IOFIX(3)),(M,IOFIX(4)),(MF,IOFIX(5)),
     2(MC,IOFIX(6)),(NOUT,IOFIX(7)),(NVER,IOFIX(8)),
     3(K,IOFIX(9)),(ITER,IOFIX(10)),(INVC,IOFIX(11)),
     4(NUMVR,IOFIX(12)),(NUMPV,IOFIX(13)),
     5(INFS,IOFIX(14)),(JT,IOFIX(15)),(LA,IOFIX(16)),
     6(ZZ(1),TPIV),(ZZ(2),TZERO),(ZZ(3),TCOST)
      C
      C      *****   INITIALIZE THE VARIABLES.   *****
      C
            DO 1340 I=1,8
              TERR(I)=0.0
              IOFIX(I+8)=0
 1340       IOFIX(I)=INFIX(I)
            DO 1308 I=1,3
 1308         ZZ(I)=TOL(I)
            TCOST=-ABS(TCOST)
            PMIX=PRM
            M2=M**2
            INFS=1
      C
      C      *****   CHECK FOR ILLEGAL INPUT.   *****
      C
            IF(N) 1304,1304,1371
 1371       IF(M-MF)1304,1304,1372
 1372       IF(MF-MC)1304,1304,1373
 1373       IF(MC)1304,1304,1374
 1374       IF(ME-M)1304,1375,1375
 1375       IF( MOD(INFLAG,4)-1)1400,1320,100
      C
      C      *****   RECORD INFORMATION ABOUT THE INITIAL BASIC FEASIBLE          *****
      C      *****   SOLUTION.  (IT EXISTS DUE TO THE SET-UP IN THE MAIN PROGRAM.)*****
      C      *****      JH(I) = J   MEANS J IS THE (I-1)ST BASIC VARIABLE         *****
      C      *****                  INCLUDING SLACKS AND ARTIFICIALS.            *****
      C      *****      KB(J) = I   MEANS VARIABLE J IS BASIC IN ROW (I-1).      *****
      C      *****      KB(J) = 0   MEANS J IS NONBASIC.                         *****
      C      *****   (RECALL THAT THE COST COEFFICIENTS HAVE BEEN INSTALLED IN    *****
      C      *****   ROW 1 OF A.)                                                *****
      C
 1400       DO 1401 I=1,M
 1401         JH(I)=0
      C
      C      *****   INITIALIZE KB AND JH TO REPRESENT THE INITIAL BASIC   *****
      C      *****   FEASIBLE SOLUTION.  THIS SOLUTION EXISTS DUE TO THE   *****
      C      *****   SET-UP ON THE MAIN PROGRAM.                           *****
      C
            DO 1402 J=1,N
              KB(J)=0
              KQ=0
              LQ=0
              DO 1403 L=MF,M
                IF(A(L,J))1404,1403,1404
 1404           KQ=LQ+1
                LQ=L
 1403           CONTINUE
      C
      C      *****   CHECK WHETHER VARIABLE J IS A CANDIDATE TO BE BASIC.   *****
```

```
2520.          C
2521.                IF(KQ-1)1402,1405,1402
2522.          1405  IA=LQ
2523.                IF(JH(IA))1402,1406,1402
2524.          1406  IF(A(IA,J)*B(IA))1402,1407,1407
2525.          C
2526.          C
2527.          C     *****   J IS A BASIC VARIABLE IN ROW (IA-1),    *****
2528.          C     *****   SET JH AND KB ACCORDINGLY.              *****
2529.          C
2530.          C
2531.          1407    JH(IA)=J
2532.                  KB(J)=IA
2533.          1402    CONTINUE
2534.          1320  CONTINUE
2535.          1100  ASSIGN 1102 TO KPIV
2536.                ASSIGN 1114 TO KJMY
2537.                IF(LA)1121,1121,1122
2538.          1121  INVC=0
2539.          1122  NUMVR=NUMVR+1
2540.                DO 1101 I=1,M
2541.                  DO 9 J=1,M
2542.          9         E(I,J)=0.
2543.          1101    CONTINUE
2544.          C
2545.          C     *****   INITIALIZE THE INVERSE AT THE IDENTITY AND STORE IT IN *****
2546.          C     *****   E.  ALSO INITIALIZE X, WHERE X(I) IS THE VALUE OF THE  *****
2547.          C     *****   BASIC VARIABLE IN ROW (I-1).                           *****
2548.          C
2549.                DO 1113 I=1,M
2550.                  E(I,I)=1.0
2551.                  X(I)=B(I)
2552.          1113    CONTINUE
2553.                DO 1110 I=MF,M
2554.                  IF(JH(I))1111,1110,1111
2555.          1111    JH(I)=12345
2556.          1110    CONTINUE
2557.                INFS=1
2558.                JT=1
2559.          1     IF(KB(JT))600,1102,600
2560.          C
2561.          C     *****   FIND MAX OVER I OF ABS(Y(I)) = TY, STORE THE    *****
2562.          C     *****   ARG MAX IN IR.                                  *****
2563.          C
2564.          1114  TY=0.
2565.                DO 1104 I=MF,M
2566.                  IF(JH(I)-12345)1104,1105,1104
2567.          1105    IF(ABS(Y(I))-TY)1104,1104,1106
2568.          1106    IR=I
2569.                  TY=ABS(Y(I))
2570.          1104    CONTINUE
2571.                IF(TY-TPIV)1107,1108,1108
2572.          1107  KB(JT)=0
2573.                GO TO 1102
2574.          C
2575.          C     *****   VARIABLE JT BECOMES BASIC IN ROW (IR-1).   *****
2576.          C
2577.          1108  JH(IR)=JT
2578.                KB(JT)=IR
2579.                GO TO 900
```

```
2580.      1102  JT=JT+1
2581.            IF(JT.GT.N)GO TO 2
2582.            GO TO 1
2583.      2     DO 1109 I=1,M
2584.              IF(JH(I)-12345)1109,1112,1109
2585.      1112    JH(I)=0
2586.      1109    CONTINUE
2587.      100   ASSIGN 705 TO NDEL
2588.            ASSIGN 1000 TO KJMY
2589.            ASSIGN 221 TO KPIV
2590.      C
2591.      C     *****   CHECK FOR INFEASIBILITIES.  NEG = 1 MEANS    *****
2592.      C     *****   X(I) .LT. 0 FOR SOME I.  JIN = 1 MEANS       *****
2593.      C     *****   X(I) .LT. 0 FOR SOME I OR JH(I) = 0 FOR      *****
2594.      C     *****   SOME I.                                      *****
2595.      C
2596.      1200  JIN=0
2597.            NEG=0
2598.            DO 1201 I=MF,M
2599.              IF(ABS(X(I))-TZERO)1202,1203,1203
2600.      1202    X(I)=0.0
2601.              GO TO 1201
2602.      1203    IF(X(I))1208,1201,1205
2603.      1205    IF(JH(I))1201,1206,1201
2604.      1208    NEG=1
2605.      1206    JIN=1
2606.      1201    CONTINUE
2607.            IF(INFS-JIN)1320,500,200
2608.      C
2609.      C     *****   A FEASIBLE SOLUTION HAS BEEN FOUND.   *****
2610.      C
2611.      200   INFS=0
2612.      201   PMIX=0.0
2613.      500   CONTINUE
2614.      C
2615.      C     *****   INITIALIZE THE PRICES.   *****
2616.      C
2617.      502   DO 503 J=1,M
2618.              P(J)=E(1,J)
2619.      503     CONTINUE
2620.            IF(INFS)501,599,501
2621.      C
2622.      C     *****   UPDATE THE PRICES.   *****
2623.      C
2624.      501   DO 504 J=1,M
2625.      504     P(J)=P(J)*PMIX
2626.            DO 505 I=MF,M
2627.              IF(X(I))506,507,507
2628.      506     DO 508 J=1,M
2629.                P(J)=P(J)+E(I,J)
2630.      508       CONTINUE
2631.              GO TO 505
2632.      507     IF(JH(I)) 505,509,505
2633.      509     DO 510 J=1,M
2634.                P(J)=P(J)-E(I,J)
2635.      510       CONTINUE
2636.      505     CONTINUE
2637.      599   CONTINUE
```

```
      C
  700 JT=0
      BB=TCOST
  701 JM=1
      C
      C     *****   SKIP THE COLUMNS ALREADY IN THE BASIS.   *****
      C
  703 IF(KB(JM))702,300,702
      C
      C     *****   FIND THE MOST NEGATIVE PRICED OUT COLUMN.   *****
      C     *****   STORE THE INDEX IN JT.                      *****
      C
  705 IF(DT-BB)708,702,702
  708 BB=DT
      JT=JM
  702 JM=JM+1
      IF(JM.GT.N)GO TO 3
      GO TO 703
    3 IF(JT)203,203,600
      C
      C     *****   ALL COSTS ARE NON-NEGATIVE...K=3 OR 4.   *****
      C
  203 K=3+INFS
      GO TO 257
      C
      C     *****   JT IS A BASIC VARIABLE.  STORE    *****
      C     *****   (ROW J OF E) * (COLUMN JT OF A)   *****
      C     *****   IN Y(J).                          *****
      C
  600 DO 610 I=1,M
  610   Y(I)=0.
      DO 605 I=1,M
        IF(A(I,JT))601,602,601
  601   DO 606 J=1,M
  606     Y(J)=Y(J)+A(I,JT)*E(J,I)
        GO TO 605
  602   CONTINUE
  605   CONTINUE
  699 GO TO KJMY,(1000,1114,1392)
      C
      C     *****   FIND WHICH ROW JT (VIA Y) WILL BECOME BASIC IN.   *****
      C
 1000 IR=0
      AA=0.0
      IA=0
      C
      C     *****   IF THERE ARE X(I) = 0, STORE MAX AMONG THESE I OF ABS(Y)   *****
      C     *****   IS AA, STORE THE ARG MAX IN IR.                           *****
      C
      DO 1050 I=MF,M
        IF(X(I))1050,1041,1050
 1041   YI=ABS(Y(I))
        IF(YI-TPIV)1050,1050,1042
 1042   IF(JH(I))1043,1044,1043
 1043   IF(IA)1050,1048,1050
 1048   IF(Y(I))1050,1050,1045
 1044   IF(IA)1045,1046,1045
 1045   IF(YI-AA)1050,1050,1047
 1046   IA=1
 1047   AA=YI
```

```
2700.                IR=I
2701.      1050      CONTINUE
2702.              IF (IR) 1099,1001,1099
2703.      1001    AA=1.0E+20
2704.      C
2705.      C       *****   THERE ARE NO X(I) = 0.  FIND THE MINIMUM PIVOT AMONG    *****
2706.      C       *****   THE POSITIVE EQUATIONS.  STORE THE MINIMUM VALUE IN     *****
2707.      C       *****   AA, WITH THE ARG MIN IN IR.                             *****
2708.      C
2709.              DO 1010 IT=MF,M
2710.                IF(Y(IT)-TPIV) 1010,1010,1002
2711.      1002      IF(X(IT))1010,1010,1003
2712.      1003      XY=X(IT)/Y(IT)
2713.                IF(XY-AA)1004,1005,1010
2714.      1005      IF(JH(IT))1010,1004,1010
2715.      1004      AA=XY
2716.                IR=IT
2717.      1010      CONTINUE
2718.              IF(NEG)1016,1099,1016
2719.      C
2720.      C       *****   FIND PIVOT AMONG NEGATIVE EQUATIONS, IN WHICH    *****
2721.      C       *****   X/Y IS LESS THAN AA, THAT HAS THE LARGEST        *****
2722.      C       *****   ABS(Y), STORE THE ARG MAX IN IR.                 *****
2723.      C
2724.      1016    BB=-TPIV
2725.              DO 1030 I=MF,M
2726.                IF(X(I))1012,1030,1030
2727.      1012      IF(Y(I)-BB)1022,1030,1030
2728.      1022      IF(Y(I)*AA-X(I)) 1024,1024,1030
2729.      1024      BB=Y(I)
2730.                IR=I
2731.      1030      CONTINUE
2732.      1099    CONTINUE
2733.      206     IF(IR)207,207,210
2734.      C
2735.      C       *****   NO PIVOT ROW HAS BEEN FOUND FOR COLUMN JT.   *****
2736.      C
2737.      207     K=5
2738.      257     IF(PMIX)201,400,201
2739.      C
2740.      C       *****   CHECK IF THE ITERATION LIMIT HAS BEEN EXCEEDED.   *****
2741.      C
2742.      210     IF(ITER-NCUT) 900,160,160
2743.      900     NUMPV=NUMPV+1
2744.              YI=-Y(IR)
2745.              Y(IR)=-1.
2746.      C
2747.      C       *****   TRANSFORM THE INVERSE TO CORRESPOND  *****
2748.      C       *****   TO JT BEING BASIC IN ROW (IR-1).     *****
2749.      C
2750.      903     DO 904 L=1,M
2751.                IF(E(IR,L))905,914,905
2752.      914       CONTINUE
2753.                GO TO 904
2754.      905       XY=E(IR,L)/YI
2755.                E(IR,L)=0
2756.                DO 906 I=1,M
2757.      906         E(I,L)=E(I,L)+XY*Y(I)
2758.      904       CONTINUE
2759.      C
```

```
2760.      C      *****   TRANSFORM X TO CORRESPOND TO   *****
2761.      C      *****   JT BEING BASIC IN ROW (IR-1).  *****
2762.      C
2763.            XY=X(IR)/YI
2764.            X(IR)=0.
2765.            DO 908 I=1,M
2766.      908     X(I)=X(I)+XY*Y(I)
2767.            Y(IR)=-YI
2768.      999   GO TO KPIV,(221,1102)
2769.      C
2770.      C     *****   ADJUST JH AND KB TO MAKE JT BASIC IN ROW (IR-1).   *****
2771.      C
2772.      221   IA=JH(IR)
2773.            IF(IA)213,213,214
2774.      214   KB(IA)=0
2775.      213   KB(JT)=IR
2776.            JH(IR)=JT
2777.            LA=0
2778.            ITER=ITER+1
2779.            INVC=INVC+1
2780.      C
2781.      C     *****   CHECK THE INVERSION FREQUENCY.   *****
2782.      C
2783.            IF(INVC-NVER)1200,1320,1200
2784.      C
2785.      C     *****   TOO MANY ITERATIONS HAVE BEEN PERFORMED.   *****
2786.      C
2787.      160   K=6
2788.      C
2789.      C     *****   STORE AX - B IN Y.   *****
2790.      C
2791.      400   ASSIGN 410 TO NDEL
2792.            DO 401 I=1,M
2793.      401     Y(I)=-B(I)
2794.            DO 402 I=1,M
2795.              JA=JH(I)
2796.              IF(JA) 403,402,403
2797.      403     DO 405 IT=1,M
2798.                IF(A(IT,JA))415,405,415
2799.      415         Y(IT)=Y(IT)+X(I)*A(IT,JA)
2800.      405       CONTINUE
2801.      402     CONTINUE
2802.      C
2803.      C     *****   FIND THE SUM AND THE MAXIMUM OF THE ERRORS IN      *****
2804.      C     *****   AX - B, STORE IN TERR(1) AND TERR(2) RESPECTIVELY.  *****
2805.      C
2806.            DO 481 I=1,M
2807.              YI=Y(I)
2808.              IF(JH(I)) 472,471,472
2809.      471     YI=YI+X(I)
2810.      472     TERR(LA+1)=TERR(LA+1)+ABS(YI)
2811.              IF(ABS(TERR(LA+2))-ABS(YI))482,481,481
2812.      482     TERR(LA+2)=YI
2813.      481     CONTINUE
2814.      C
2815.      C     *****   STORE P TIMES THE BASIS AT DT.   *****
2816.      C
```

```
2820.      C
2821.      C     *****   FIND THE SUM AND THE MAXIMUM OF THE ERRORS IN       *****
2822.      C     *****   PA - 0, STORE IN TERR(3) AND TERR(4) RESPECTIVELY.  *****
2823.      C
2824.      410   TERR(LA+3)=TERR(LA+3)+ABS(DT)
2825.            IF(ABS(TERR(LA+4)) - ABS(DT)) 413,411,411
2826.      413   TERR(LA+4)=DT
2827.      411   I=I+1
2828.            IF(I.GT.M)GO TO 5
2829.            GO TO 4
2830.      5     IF(LA) 193,191,193
2831.      191   LA=4
2832.            IF(INFLAG-4)1320,193,193
2833.      193   IF(K-5)1392,194,1392
2834.      194   ASSIGN 1392 TO KJMY
2835.            GO TO 600
2836.      1304  K=7
2837.      C
2838.      C     *****   SET EXIT VALUES FOR RETURN TO THE MAIN PROGRAM.    *****
2839.      C
2840.      1392  DO 1309 I=1,8
2841.      1309    ERR(I)=TERR(I)
2842.            DO 1329 I=1,7
2843.      1329    KOUT(I)=IOFIX(I+8)
2844.            RETURN
2845.      C
2846.      C     *****   TEST JM FOR ENTERRING THE BASIS.  PRICE OUT    *****
2847.      C     *****   JM AND STORE THIS VALUE IN DT.  (I.E. STORE    *****
2848.      C     *****   P * (COLUMN JM OF A) IN DT.)                   *****
2849.      C
2850.      300   DT=0.
2851.      301   DO 303 MM=1,M
2852.              IF(A(MM,JM))304,303,304
2853.      304     DT=DT+P(MM)*A(MM,JM)
2854.      303     CONTINUE
2855.      399   GO TO NDEL,(410,705)
2856.            END
2857.      C
2858.      C
2859.      C
2860.      C
2861.      C
2862.            SUBROUTINE DUAL (A,B,C,JH,X,Y,M,N,SIGN)
2863.      C
2864.      C
2865.      C     ***************************************************************************
2866.      C     *       THIS SUBROUTINE INITIALIZES THE DUAL SIMPLEX CODE.                *
2867.      C     ***************************************************************************
2868.      C
2869.      C
2870.            DIMENSION A(61,181),AA(11284),INORDR(62),L1(61),L2(120)
2871.            DIMENSION B(M),C(N),X(N),Y(M),JH(M)
2872.            INTEGER PROTO(120),DVAR1(61),DVAR2(120)
2873.            LOGICAL SIGN
2874.            MPLUS1=M+1
2875.            NN=N-M
2876.            IF(SIGN)NN=N
2877.            MN=MPLUS1*(NN+1)
2878.            CALL INPUT (A,B,C,JH,X,Y,M,N,SIGN,MPLUS1,NN,MN,AA,INORDR,
2879.           1DVAR1,DVAR2,PROTO,L1,L2)
```

```
      RETURN
      END
C
C
C
C
      SUBROUTINE INPUT (A,B,C,JH,X,Y,M,N,SIGN,MPLUS1,NN,MN,AA,INORDR,
     1DVAR1,DVAR2,PROTO,L1,L2)
C
C
C*****************************************************************
C*       THIS SUBROUTINE PREPARES INPUT FOR THE MAIN PROGRAM OF THE  *
C*       DUAL SIMPLEX CODE.                                          *
C*****************************************************************
C
C
      DIMENSION A(61,181),B(M),C(N),Y(M),JH(M),AA(11284),INORDR(MPLUS1)
      DIMENSION X(N)
      DIMENSION L1(61),L2(120)
      INTEGER PROTO(120),DVAR1(61),DVAR2(120),FALL
      LOGICAL SIGN
1000  FORMAT(' THE PROGRAM IS INFEASIBLE.')
      IF(SIGN) GO TO 200
C
C*****   GO TO 200 IF THE CONSTRAINTS ARE 'GREATER THAN OR EQUAL'.   *****
C
      DO 100 K=1,M
100     INORDR(K)=JH(K)
C
C*****   SORT INORDR INTO ASCENDING ORDER.   *****
C
      MLESS1=M-1
      DO 120 K=1,MLESS1
        HOLD=INORDR(K)
        KPLUS1=K+1
        DO 120 J=KPLUS1,M
          IF(INORDR(J).GE.HOLD) GO TO 120
          HOLD=INORDR(J)
          INORDR(J)=INORDR(K)
          INORDR(K)=HOLD
120       CONTINUE
      INORDR(MPLUS1)=N+1
C
C*****   THE NEXT SEQUENCE OF STATEMENTS FORM THE AA MATRIX,  *****
C*****   WHICH WILL BE INPUT FOR SUBROUTINE DUSEX.            *****
C
      L=1
      DO 140 J=1,N
        IF(J.LT.INORDR(L)) GO TO 130
        L=L+1
        GO TO 140
130     K=MPLUS1*(J-L+1)+1
        AA(K)=C(J)
        PROTO(J-L+1)=J
C
C*****   PROTO IS AN ARRAY CONTAINING THE INDICES OF   *****
C*****   THE NONBASIC VARIABLES.                       *****
C
        DVAR2(J-L+1)=0
```

```
C
C     *****   DVAR2 FLAGS VARIABLES WHICH MAY NOT BE BASIC    *****
C     *****   IN THE DUAL.                                    *****
C
      DO 135 I=1,M
        AA(K+I)=A(I,J)
135   CONTINUE
140   DO 150 I=1,M
        AA(I+1)=B(I)
150     DVAR1(I)=I
C
C     *****   IT MAY BE DESIRABLE TO PASS AN INITIAL VALUE OF Z    *****
C     *****   IN THE CALLING SEQUENCE.  THE STATEMENT BELOW ARBI-  *****
C     *****   TRARILY SETS Z TO 0.0 TO BEGIN.                      *****
C
      AA(1)=0.0
      CALL DUSEX(AA,MN,1,MPLUS1,NN,M,FALL,.TRUE.,PROTO,DVAR2,NN,
     1JH,DVAR1,M,L1,M,L2,NN)
C
C     *****   RETRIEVE INFORMATION FROM THE DUAL SIMPLEX ROUTINE.   *****
C
      DO 165 J=1,NN
        K=PROTO(J)
        L=J*MPLUS1+1
        C(K)=AA(L)
        DO 165 I=1,M
          A(I,K)=AA(L+I)
165     CONTINUE
      DO 168 I=1,M
        B(I)=AA(I+1)
        K=JH(I)
        DO 167 J=1,M
          A(J,K)=0.0
        C(K)=0.0
        A(I,K)=1.0
168     CONTINUE
      IF(FALL.EQ.2) GO TO 190
C
C     *****   FIND THE SOLUTION FOR THE PRIMAL PROBLEM AND STORE IT IN X.  *****
C
      DO 170 K=1,N
170     X(K)=0.0
      DO 175 K=1,M
        L=JH(K)
        X(L)=AA(K+1)
175     Y(K)=0.0
C
C     *****   FIND THE SOLUTION TO THE DUAL PROBLEM AND STORE IT IN Y.   *****
C
179   DO 180 K=1,NN
        L=DVAR2(K)
        IF(L.NE.0) Y(L)=AA(K+MPLUS1+1)
180     CONTINUE
      RETURN
190   WRITE(6,1000)
      RETURN
C
C     *****   THE CONSTRAINTS ARE 'GREATER THAN OR EQUAL'.  THE    *****
C     *****   NEXT SEQUENCE OF STATEMENTS FORM THE AA MATRIX,      *****
C     *****   WHICH WILL BE INPUT FOR SUBROUTINE DUSEX.  SET THE   *****
```

```
C      *****   INITIAL VALUE OF Z TO 0.0.                                  *****
C
200   AA(1)=0.0
      DO 240 J=1,N
        K=MPLUS1*J+1
        AA(K)=C(J)
        DO 230 I=1,M
230       AA(K+I)=-A(I,J)
C
C      *****   PROTO IS AN ARRAY CONTAINING THE INDICES OF    *****
C      *****   THE NONBASIC VARIABLES.                        *****
C
        PROTO(J)=J
C
C      *****   DVAR2 FLAGS VARIABLES WHICH MAY NOT BE BASIC   *****
C      *****   IN THE DUAL.                                   *****
C
        DVAR2(J)=0
240     X(J)=0.0
      DO 250 K=1,M
        AA(K+1)=-B(K)
        DVAR1(K)=K
        JH(K)=N+K
250     Y(K)=0.0
      CALL DUSEX (AA,MN,1,MPLUS1,NN,M,FALL,.TRUE.,PROTO,DVAR2,NN,
     1JH,DVAR1,M,L1,M,L2,NN)
C
C      *****   RETRIEVE INFORMATION FROM THE DUAL SIMPLEX ROUTINE.   *****
C
      L=1
      DO 260 J=1,N
        L=L+MPLUS1
        C(J)=AA(L)
        DO 260 I=1,M
          A(I,J)=-AA(L+I)
260       CONTINUE
      DO 265 I=1,M
265     B(I)=-AA(I+1)
      IF(FALL.EQ.2) GO TO 190
C
C      *****   FIND THE SOLUTION FOR THE PRIMAL PROBLEM (NOT          *****
C      *****   INCLUDING SLACKS) AND STORE IT IN X.                   *****
C
      DO 270 K=1,N
        L=JH(K)
        IF(L.LE.N) X(L)=AA(K+1)
270     CONTINUE
      GO TO 179
      END
C
C
C
C
      SUBROUTINE DUSEX (A,J1,ZSCHR,SSCHR,N,M,FALL,W,PROTO1,DVAR2,
     1IPROT1,PROTO2,DVAR1,IPROT2,L1,JL1,L2,JL2)
C
C      ***********************************************************************
C      *       THIS IS THE MAIN BODY OF THE DUAL SIMPLEX ROUTINE FOR DETER-  *
```

```
C     *       MINING THE MINIMUM OF A LINEAR PROGRAM BY MEANS OF THE DUAL        *
C     *       SIMPLEX METHOD.  THE TABLEAU IS ASSUMED TO BE DUALLY FEASIBLE.     *
C     *       DUSEX USES THE SUBROUTINES MP2,MP3,MP5.--FROM NUMERICAL METHODS    *
C     *       OF MATHEMATICAL OPTIMIZATION, KUNZI,TZSCHACH AND ZEHNDER.  IT      *
C     *       ASSUMES THAT (1) ALL CONSTRAINTS ARE INEQUALITIES, AND (2) ALL     *
C     *       COEFFICIENTS OF THE OBJECTIVE FUNCTION TO BE MINIMIZED ARE         *
C     *       NONNEGATIVE.                                                       *
C     ************************************************************************
C
      DIMENSION A(11284),L1(JL1),L2(120)
      INTEGER PROTO1(120),PROTO2(M),FALL,ZSCHR,SSCHR,V,ZNR
      INTEGER DVAR1(M),DVAR2(120)
      LOGICAL W
C
C     *****   A IS THE SIMPLEX TABLEAU.  J1 IS THE DIMENSION OF A,    *****
C     *****   I.E. J1=(M+1)*(N+1) WHERE N IS THE NUMBER OF VARIABLES  *****
C     *****   IN THE LINEAR PROGRAM AND M IS THE NUMBER OF CON-       *****
C     *****   STRAINTS IN THE LINEAR PROGRAM.  FALL IS A CODE SPECI-  *****
C     *****   FYING THE OUTCOME OF THE PROBLEM.  PROTO1 IS AN ARRAY   *****
C     *****   CONTAINING THE INDICES OF THE NONBASIC VARIABLES.       *****
C     *****   PROTO2 IS AN ARRAY CONTAINING THE INDICES OF THE BASIC  *****
C     *****   VARIABLES.  L1 IS AN ARRAY CONTAINING THE INDICES OF    *****
C     *****   THE COLUMNS ADMISSABLE FOR AN EXCHANGE.  JL1 IS THE     *****
C     *****   DIMENSION OF L1.  L10 IS THE NUMBER OF COLUMNS ADMIS-   *****
C     *****   SABLE FOR AN EXCHANGE.  L2 IS AN ARRAY CONTAINING THE   *****
C     *****   INDICES OF THE ROWS ADMISSABLE FOR AN EXCHANGE.  JL2    *****
C     *****   IS THE DIMENSION OF L2.  L20 IS THE NUMBER OF ROWS      *****
C     *****   ADMISSABLE FOR AN EXCHANGE.                            *****
C
      DO 1 K=1,M
    1   L1(K)=K
      L10=M
      DO 2 I=1,N
    2   L2(I)=I
      L20=N
      V=1
C
C     *****   W=.TRUE. MEANS A BASIS IS GIVEN, OTHERWISE ONE    *****
C     *****   MUST BE INITIALIZED.                              *****
C
      IF(W) GO TO 100
      DO 3 K=1,N
    3   PROTO1(K)=K
      DO 4 I=1,M
    4   PROTO1(I)=I+N
C
C     *****   THE SIX STATEMENTS BELOW EFFECTIVELY TRANSPOSE THE   *****
C     *****   TABLEAU A FOR THE PURPOSE OF SEARCHING FOR THE MIN-  *****
C     *****   IMUM ELEMENT OF THE CONSTANT COLUMN.  THE GENERALITY *****
C     *****   OF THE SUBROUTINES MP2,MP3 AND MP5 MAKE THE TRANS-   *****
C     *****   POSITION NECESSARY.                                  *****
C
  100 I=ZSCHR
      ZSCHR=SSCHR
      SSCHR=I
      I=M
      M=N
      N=I
      CALL MP5 (A,J1,L1,L10,JL1,Q1,KP,0,ZSCHR,SSCHR)
```

```
C
C     *****   A BRANCH TO 101 INDICATES THE SOLUTION   *****
C     *****   IS NOT YET OPTIMAL.                      *****
C
      IF(Q1.LT.0.) GO TO 101
      FALL=0
      GO TO 300
  101 CALL MP2(A,J1,L2,L20,JL2,IP,ZSCHR,SSCHR,KP,Q1,N,V)
      IF(IP.NE.0) GO TO 200
      FALL=2
      GO TO 300
C
C     *****   THE TWELVE STATEMENTS BELOW EFFECTIVELY TRANSPOSE   *****
C     *****   THE TABLEAU A BACK TO ITS ORIGINAL DIMENSIONS       *****
C     *****   BEFORE THE PIVOTING STEP IS BEGUN.                 *****
C
  200 I=ZSCHR
      ZSCHR=SSCHR
      SSCHR=I
      I=M
      M=N
      N=I
      I=KP
      KP=IP
      IP=I
      I=PROTO1(KP)
      PROTO1(KP)=PROTO2(IP)
      PROTO2(IP)=I
      I=DVAR2(KP)
      DVAR2(KP)=DVAR1(IP)
      DVAR1(IP)=I
      CALL MP3 (A,J1,0,M,0,N,IP,KP,ZSCHR,SSCHR,1,1)
      GO TO 100
C
C     *****   AN OPTIMAL SOLUTION HAS BEEN FOUND.  RESTORE   *****
C     *****   THE VARIABLES TO THEIR ORIGINAL VALUES.        *****
C
  300 I=ZSCHR
      ZSCHR=SSCHR
      SSCHR=I
      I=M
      M=N
      N=I
      A(1)=-A(1)
      RETURN
      END
C
C
C
C
      SUBROUTINE MP2(A,J1,L2,L20,JL2,IP,ZSCHR,SSCHR,KP,Q1,N,IV)
C
C
C*************************************************************************
C*       SUBROUTINE MP2 DETERMINES THE MINIMUM OF A LIST OF QUOTIENTS.   *
C*       THAT IS, THE PIVOT COLUMN IS SELECTED BY MP2.  IP IS THE PIVOT  *
C*       COLUMN AND Q1 IS THE MINIMUM.                                   *
C*************************************************************************
C
```

```
      C
      DIMENSION A(J1),L2(JL2)
      INTEGER ZSCHR,SSCHR,Z
      C
      C     *****   MP3 DETERMINES THE MINIMUM OF ALL THOSE       *****
      C     *****   V*A(I*ZSCHR)/A(I*ZSCHR+KP*SSCHR), FOR WHICH   *****
      C     *****   THE A(I*ZSCHR) ARE NONNEGATIVE AND THE        *****
      C     *****   V*A(I*ZSCHR+KP*SSCHR) ARE POSITIVE.  DEGEN-   *****
      C     *****   ERACY IS TAKEN INTO CONSIDERATION.            *****
      C
 1000 V=IV
      IP=0
      IF(L20.LT.1) RETURN
      DO 1 I=1,L20
        KH=L2(I)*ZSCHR+1
        KH1=KH+KP*SSCHR
        IF(V*A(KH1).LT.0) GO TO 2
    1   CONTINUE
      RETURN
    2 Q1=V*A(KH)/A(KH1)
      IP=L2(I)
      Z=I+1
      IF(Z.GT.L20) RETURN
      DO 3 I=Z,L20
        KH=L2(I)*ZSCHR+1
        KH1=KH+KP*SSCHR
        IF(V*A(KH1).GE.0.)GO TO 3
        Q=V*A(KH)/A(KH1)
        IF(Q.LE.Q1) GO TO 4
        IP=L2(I)
        Q1=Q
        GO TO 3
    4   IF(Q.NE.Q1) GO TO 3
      C
      C     *****   HERE IT IS DETERMINED WHICH OF TWO ROWS WITH EQUAL   *****
      C     *****   QUOTIENTS QUALIFIES AS PIVOT ROW.                   *****
      C
        I0=L2(I)
        DO 5 K=1,N
          KH0=IP*ZSCHR+K*SSCHR+1
          KH2=IP*ZSCHR+KP*SSCHR+1
          KH=I0*ZSCHR+K*SSCHR+1
          QP=V*A(KH0)/A(KH2)
          Q0=V*A(KH)/A(KH1)
          IF(QP.GT.Q0) GO TO 3
          IF(Q0.GT.QP) GO TO 6
    5     CONTINUE
    6   IP=I0
    3   CONTINUE
      RETURN
      END
      C
      C
      C
      C
      C
      SUBROUTINE MP3(A,J1,I0,I1,K0,K1,IP,KP,ZSCHR,SSCHR,P1,P2)
      C
      C
      C     ***********************************************************
```

```
C      *       SUBROUTINE MP3 EXCHANGES A BASIC AND A NONBASIC VARIABLE IN       *
C      *       THE TABLEAU.                                                      *
C      ************************************************************************
C
C
      DIMENSION A(J1)
      INTEGER ZSCHR,SSCHR,P1,P2
C
C      *****    I0 AND I1 GIVE THE ROWS AND K0 AND K1 THE COLUMNS TO    *****
C      *****    WHICH THE TRANSFORMATION IS TO APPLY.  P1 AND P2        *****
C      *****    INDICATE WHETHER THE PIVOT ROW OR COLUMN HAS ALREADY    *****
C      *****    BEEN TRANSFORMED.                                       *****
C
      KH=IP*ZSCHR+KP*SSCHR+1
      PIV=1./A(KH)
      II0=I0+1
      II1=I1+1
      KK0=K0+1
      KK1=K1+1
      DO 1 II=II0,II1
        I=II-1
        IF(I.EQ.IP) GO TO 1
        KH0=I*ZSCHR+KP*SSCHR+1
        IF(P2.EQ.1) A(KH0)=-A(KH0)*PIV
        DO 2 KK=KK0,KK1
          K=KK-1
          IF(K.EQ.KP) GO TO 2
          KH1=I*ZSCHR+K*SSCHR+1
          KH2=IP*ZSCHR+K*SSCHR+1
          A(KH1)=A(KH1)+A(KH2)*A(KH0)
    2     CONTINUE
    1   CONTINUE
      IF(P1.NE.1) GO TO 4
      DO 5 KK=KK0,KK1
        K=KK-1
        KH2=IP*ZSCHR+K*SSCHR+1
    5   IF(K.NE.KP) A(KH2)=A(KH2)*PIV
    4 IF(P2.EQ.1) A(KH)=PIV
      RETURN
      END
C
C
C
C
      SUBROUTINE MP5(A,J1,L1,L10,JL1,Q1,KP,ZNR,ZSCHR,SSCHR)
C
C      ************************************************************************
C      *      SUBROUTINE MP5 FINDS THE MINIMUM VALUE IN THE CONSTANT COLUMN    *
C      *      AND STORES THE RESULT IN Q1.                                     *
C      ************************************************************************
C
C
      INTEGER ZNR,ZSCHR,SSCHR
      DIMENSION A(J1),L1(JL1)
C
C      *****    THE ENTRIES IN THE CONSTANT COLUMN ARE      *****
C      *****    A(ZNR*ZSCHR+L1(K)*SSCHR) FOR K=1,....,L10.  *****
C      *****    KP IS THE INDEX K WHICH POINTS TO THE       *****
```

```
C      *****   MINIMUM Q1.                                          *****
C
       KP=L1(1)
       KH=ZNR*ZSCHR+1
       KH0=KH+L1(1)*SSCHR
       Q1=A(KH0)
       IF(L10.LT.2) RETURN
       DO 1 K=2,L10
         KH0=KH+L1(K)*SSCHR
         IF(A(KH0).GE.Q1) GO TO 1
         Q1=A(KH0)
         KP=L1(K)
1        CONTINUE
       RETURN
       END
C
C
C
C
C
       SUBROUTINE PART12
C
C
C      ***********************************************************************
C      *      OBTAIN NEW LOWER AND UPPER BOUNDS FOR THE VARIABLES IN GROUPS  *
C      *      3 AND 4 GIVEN X(1),...,X(MIN(J-1,N3)) AND ZB.                  *
C      ***********************************************************************
C
C
       DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
      1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
      2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
      3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
      4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
      6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
       COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
      XINDEXG,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
      XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
      XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
      XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
      XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
       NJ=J-1
       IF(NJ.GT.N3)NJ=N3
       T=D1(J,N+1)-Y1(J)*ZB
       TT=D2(J,N+1)+Y2(J)*ZB
       IF(NJ.LE.0)GO TO 53
C
C      *****   T=U'(J);  TT=L'(J) WHERE U'(-) AND L'(-) ARE DEFINED   *****
C      *****   IN REFERENCE 2, PP. 43-44.                               *****
C
       DO 10 K=1,NJ
         T=T+D1(J,K)*X(K)
         TT=TT+D2(J,K)*X(K)
10       CONTINUE
53     CONTINUE
C
C      *****   FIND THE NEW UPPER BOUND AND STORE IT IN ULP(J).   *****
C
       IF(T.LT.0.0)GO TO 11
13     ULP(J)=AINT(T)
```

```
      GO TO 12
   11 IF(T.EQ.AINT(T)) GO TO 13
      ULP(J)=AINT(T)-1.0
   12 CONTINUE
C
C     *****   FIND THE NEW LOWER BOUND AND STORE IT IN ALLP(J).   *****
C
      IF(TT.GT.0.0) GO TO 15
   17 ALLP(J)=AINT(TT)
      GO TO 16
   15 IF(TT.EQ.AINT(TT))GO TO 17
      ALLP(J)=AINT(TT)+1.0
   16 RETURN
      END
C
C
C
C
      SUBROUTINE PART13
C
C
C     ***************************************************************
C     *      IF IT EXISTS, OBTAIN AN INITIAL ELIGIBLE PARTIAL SOLUTION FOR      *
C     *      THE GROUP 3 VARIABLES, GIVEN X(1),....X(N2) AND ZB.                *
C     ***************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     6XPRIME(61),Y(61),JR(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,N8,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      LL=N2+1
      J=N2+1
C
C     *****   STEP 8(REFERENCE 2, P. 44).   *****
C
  414 CONTINUE
      CALL PART12
C
C     *****   FIND THE BEST AVAILABLE UPPER BOUND, STORE IT IN UMIN(J).   *****
C
      IF(UEP(J)-ULP(J))401,401,402
  401 UMIN(J)=UEP(J)
      GO TO 403
  402 UMIN(J)=ULP(J)
C
C     *****   FIND THE BEST AVAILABLE LOWER BOUND, STORE IT IN ALMAX(J).   *****
C
  403 IF(ALEP(J)-ALLP(J))404,404,405
  404 ALMAX(J)=ALLP(J)
      GO TO 406
```

```
      405   ALMAX(J)=ALEP(J)
      406   IF(UMIN(J)-ALMAX(J))407,408,408
            C
            C     *****   STEP D(REFERENCE 2, P. 45).  FIND A CANDIDATE SOLUTION    *****
            C     *****   FOR THE GROUP 3 VARIABLES AND STORE IT IN X(J) AND        *****
            C     *****   XPRIME(J).  DELTA(J)=1.0 MEANS X(J) IS AT ITS UPPER       *****
            C     *****   BOUND.                                                    *****
            C
      408   IF(ALMAX(J).LE.XB(J).AND.XB(J).LE.UMIN(J))GO TO 409
            IF(XB(J).GT.UMIN(J)) GO TO 412
            X(J)=ALMAX(J)
            DELTA(J)=0.0
            GO TO 411
      409   X(J)=XB(J)
            DELTA(J)=0.0
            GO TO 411
      412   X(J)=UMIN(J)
            DELTA(J)=1.0
      411   XPRIME(J)=X(J)
            IF(J.EQ.N3)GO TO 413
            J=J+1
            GO TO 414
            C
            C     *****   ARRIVAL AT 413 MEANS AN INITIAL ELIGIBLE          *****
            C     *****   PARTIAL SOLUTION HAS BEEN FOUND.                  *****
            C
      413   CONTINUE
            SOLN=1.0
            RETURN
            C
            C     *****   ARRIVAL AT 415 MEANS NO SOLUTION FOR THE GROUP    *****
            C     *****   3 VARIABLES EXISTS.                               *****
            C
      415   CONTINUE
            SOLN=0.0
            RETURN
            C
            C     *****   STEP F(REFERENCE 2, P. 45).    *****
            C
      407   IF(J.EQ.LL)GO TO 415
            J=J-1
            C
            C     *****   STEP G(REFERENCE 2, P. 45).    *****
            C
            IF(DELTA(J).EQ.1.0) GO TO 416
            C
            C     *****   STEP H(REFERENCE 2, P. 45).    *****
            C
            IF(X(J).EQ.UMIN(J)) GO TO 417
            X(J)=X(J)+1.0
            J=J+1
            GO TO 414
      417   X(J)=XPRIME(J)
            DELTA(J)=1.0
            C
            C     *****   STEP I(REFERENCE 2, P. 46).    *****
            C
      416   IF(X(J).EQ.ALMAX(J))GO TO 407
```

```
      GO TO 414
      END
C
C
C
C
      SUBROUTINE PART14
C
C     ******************************************************************
C     *      IF ONE EXISTS, OBTAIN A NEW ELIGIBLE PARTIAL SOLUTION FOR THE      *
C     *      GROUP 3 VARIABLES, GIVEN X(1),...,X(N2) AND ZB.                    *
C     ******************************************************************
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(51),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      LL=N2+1
C
C     *****   STEP A(REFERENCE 2, P. 46).   *****
C
      J=N3
C
C     *****   STEP B(REFERENCE 2, P. 46).   *****
C
426   IF(DELTA(J).NE.0.0)GO TO 421
C
C     *****   STEP C(REFERENCE 2, P. 46).   *****
C
      IF(X(J).EQ.UMIN(J))GO TO 422
      X(J)=X(J)+1.0
      GO TO 423
422   X(J)=XPRIME(J)
      DELTA(J)=1.0
C
C     *****   STEP D(REFERENCE 2, P. 46).   *****
C
421   IF(X(J).GT.ALMAX(J))GO TO 424
C
C     *****   STEP E(REFERENCE 2, P. 46).   *****
C
      IF(J.EQ.LL)GO TO 425
      J=J-1
      GO TO 426
```

```
      RETURN
424   X(J)=X(J)-1.0
C
C     *****   STEP F(REFERENCE 2, PP. 46-47).   *****
C
423   IF(J.EQ.N3)GO TO 427
      J=J+1
C
C     *****   STEP G(REFERENCE 2, P. 47).  RESET THE BOUNDS.   *****
C
      CALL PART12
      IF(UEP(J).LT.ULP(J))GO TO 428
      UMIN(J)=ULP(J)
      GO TO 429
428   UMIN(J)=UEP(J)
429   IF(ALEP(J).GT.ALLP(J))GO TO 431
      ALMAX(J)=ALLP(J)
      GO TO 432
431   ALMAX(J)=ALEP(J)
C
C     *****   STEP H(REFERENCE 2, P. 47).   *****
C
432   IF(UMIN(J).GE.ALMAX(J))GO TO 433
      J=J-1
      GO TO 426
C
C     *****   STEP I(REFERENCE 2, P. 47).  FIND A NEW CAN- *****
C     *****   DIDATE SOLUTION AND STORE IT IN XPRIME AND X.*****
C
433   IF(ALMAX(J).LE.XB(J).AND.XB(J).LE.UMIN(J)) GO TO 434
      IF(XB(J).GT.UMIN(J))GO TO 435
      X(J)=ALMAX(J)
      DELTA(J)=0.0
      GO TO 436
434   X(J)=XB(J)
      DELTA(J)=0.0
      GO TO 436
435   X(J)=UMIN(J)
      DELTA(J)=1.0
436   XPRIME(J)=X(J)
      GO TO 423
C
C     *****   ARRIVAL AT 427 MEANS A NEW ELIGIBLE PARTIAL SOLUTION   *****
C     *****   HAS BEEN FOUND.                                        *****
C
427   CONTINUE
      SOLN=1.0
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PART15
C
```

```
      C
      C
       DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     6XPRIME(61),Y(61),JP(61),XF(61),KN(61),Y1(61),Y2(61)
       COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
      C
      C      *****   STEP A(REFERENCE 2, P. 49).   *****
      C
       LML=N+1
       JJ=NB+1
       DO 10 I=1,NB
         SUM=0.0
         DO 11 J=1,N
           K=LPERM(J)
           SUM=SUM+A(I,J)*XX(I,K)
11         CONTINUE
         SUM=B(I)-SUM
         V(I,LML)=B(I)/SUM
         DO 12 J=1,N
           V(I,J)=-A(I,J)/SUM
12         CONTINUE
10       CONTINUE
       SUM=ZX-ZF
       JJ=N+1
       V(JJ,LML)=-ZF/SUM
       DO 13 J=1,N
         V(JJ,J)=C(J)/SUM
13       CONTINUE
       JJ=NB+1
       NBAR=0
       DO 250 I=1,JJ
250      IPERM(I)=I
       DO 251 J=1,LML
251      JPERM(J)=J
       JJ=NB+1
       MM=N-N3
       NN=1
       INDEXQ(1)=1
       INDEXQ(2)=N-N3+1
```

```
3660.                IPERM(I)=I+1
3661.  503         CONTINUE
3662.              IPERM(1)=N+1
3663.              JJJ=JPERM(1)
3664.        C
3665.        C     *****   STEP C(REFERENCE 2, P. 50), FIND NBAR.   *****
3666.        C
3667.              DO 503 I=1,JJ
3668.                III=IPERM(I)
3669.                IF(V(III,JJJ).LE.0.0) GO TO 504
3670.                KK=1
3671.                DO 505 J=2,MM
3672.                  LML=JPERM(J)
3673.                  IF(V(III,LML).GE.0.0)KK=KK+1
3674.  505           CONTINUE
3675.                IF(KK.LT.NN)GO TO 503
3676.                NN=KK
3677.                LL=I
3678.                GO TO 503
3679.  504           NBAR=NBAR+1
3680.  503         CONTINUE
3681.        C
3682.        C     *****   PUT ROW LL IN POSITION (NBAR+1).   *****
3683.        C
3684.              IH=NN
3685.              IF(NBAR.EQ.JJ)GO TO 571
3686.              KKK=IPERM(NBAR+1)
3687.              IPERM(NBAR+1)=IPERM(LL)
3688.              IPERM(LL)=KKK
3689.              NN=1
3690.              LL=NBAR+1
3691.              SORT=1.0
3692.        C
3693.        C     *****   SORT THE COLUMNS OF W SO THAT THE NONNEGATIVE   *****
3694.        C     *****   ENTRIES OF (NBAR+1) COME FIRST.                 *****
3695.        C
3696.              CALL SRT
3697.        C
3698.        C     *****   PUT ROW LL IN POSITION 1.   *****
3699.        C
3700.              JJ=NP+1
3701.              NE=0
3702.              LL=0
3703.              JK=JPERM(1)
3704.              KKK=IPERM(NBAR+1)
3705.              DO 506 I=1,JJ
3706.                III=IPERM(I)
3707.                IF(V(III,JK).GT.0.0)GO TO 506
3708.                KK=0
3709.                DO 508 J=1,MM
3710.                  JJJ=JPERM(J)
3711.                  IF(V(III,JJJ).LE.0.0.AND.V(KKK,JJJ).GE.0.0)KK=KK+1
3712.  508           CONTINUE
3713.                IF(KK.LT.NE)GO TO 506
3714.                NE=KK
3715.                LL=I
3716.  506           CONTINUE
3717.              IF(LL.EQ.0)GO TO 532
3718.              GO TO 507
3719.  531         NE=0
```

```
      JJ=NB+1
      DO 509 I=1,JJ
        III=IPERM(I)
        KK=0
        DO 510 J=1,MM
          JJJ=JPERM(J)
          IF(V(III,JJJ).LE.0.0)KK=KK+1
510       CONTINUE
        IF(KK.LT.NE)GO TO 509
        NE=KK
        LL=I
509     CONTINUE
507   KKK=IPERM(1)
      IPERM(1)=IPERM(LL)
      IPERM(LL)=KKK
      LL=1
      SORT=2.0
C
C     *****   SORT THE COLUMNS OF W SO THAT THE NONPOSITIVE   *****
C     *****   ENTRIES OF ROW 1 COME FIRST WITHOUT VIOLATING   *****
C     *****   THE PREVIOUS SORTING.                           *****
C
      CALL SRT
C
C     *****   PUT ROW LL IN POSITION K3.   *****
C
      JK=JPERM(1)
      JW=NB+1
      IF(NBAR.EQ.1)GO TO 533
      DO 513 K3=2,NBAR
        LK=-1
        DO 514 I=K3,JW
          III=IPERM(I)
          IF(V(III,JK).GT.0.0)GO TO 514
          KK=0
          DO 515 J=1,NF
            JJJ=JPERM(J)
            IF(V(III,JJJ).LE.0.0)KK=KK+1
515         CONTINUE
          IF(KK.LT.LK)GO TO 514
          LL=I
          LK=KK
514       CONTINUE
        NE=LK
        KKK=IPERM(K3)
        IPERM(K3)=IPERM(LL)
        IPERM(LL)=KKK
        LL=K3
        SORT=2.0
C
C     *****   SORT THE COLUMNS OF W SO THAT THE NONPOSITIVE   *****
C     *****   ENTRIES OF ROW K3 PRECEDED THE POSITIVE ENTRIES *****
C     *****   WITHOUT VIOLATING THE PREVIOUS SORTING.         *****
C
        CALL SRT
513     CONTINUE
      GO TO 533
C
C     *****   PUT ROW LL IN POSITION (NBAR+2).   *****
C
```

```
      532   NE=IH
      533   CONTINUE
            IF(NBAR.GE.NB)GO TO 534
            K=NBAR+2
            MM=NB+1
            LK=-1
            DO 516 I=K,MM
              III=IPERM(I)
              KK=0
              DO 517 J=1,NE
                JJJ=JPERM(J)
                IF(V(III,JJJ).GE.0.0)KK=KK+1
      517       CONTINUE
              IF(KK.LT.LK)GO TO 516
              LK=KK
              LL=I
      516     CONTINUE
            NE=LK
            KKK=IPERM(K)
            IPERM(K)=IPERM(LL)
            IPERM(LL)=KKK
            LL=K
      C
      C     *****   SORT THE COLUMNS OF W SO THAT THE NONNEGATIVE    *****
      C     *****   ENTRIES OF ROW (NBAR+2) COME FIRST WITHOUT       *****
      C     *****   VIOLATING THE PREVIOUS SORTING.                  *****
      C
            SORT=1.0
            CALL SRT
      C
      C     *****   PUT ROW LL IN POSITION (NBAR+LT+1).      *****
      C
            IF(NBAR.GT.(NB-2))GO TO 534
            MM1=NB-NBAR
            DO 518 LT=2,MM1
              LK=-1
              JJ=NB+1
              LML=NBAR+LT+1
              DO 519 I=LML,JJ
                III=IPERM(I)
                KK=0
                DO 520 J=1,NE
                  JJJ=JPERM(J)
                  IF(V(III,JJJ).LE.0.0)KK=KK+1
      520         CONTINUE
                IF(KK.LT.LK)GO TO 519
                LL=I
                LK=KK
      519       CONTINUE
              NE=LK
              KKK=IPERM(NBAR+LT+1)
              IPERM(NBAR+LT+1)=IPERM(LL)
              IPERM(LL)=KKK
              LL=NBAR+LT+1
              SORT=1.0
      C
      C     *****   SORT THE COLUMNS OF W SO THAT THE NONNEGATIVE    *****
      C     *****   ENTRIES OF ROW (NBAR+LT+1) COME FIRST WITHOUT    *****
      C     *****   VIOLATING THE PREVIOUS SORTING.                  *****
      C
```

```
3840.                CALL SRT
3841.      518       CONTINUE
3842.      C
3843.      C         *****   IPERM AND JPERM ORDER ROWS AND COLUMNS     *****
3844.      C         *****   RESPECTIVELY SO THAT V(IPERM(I),JPERM(J)) *****
3845.      C         *****   =W(I,J) IN THE NEW ORDERING.  FOR A MORE  *****
3846.      C         *****   COMPLETE DESCRIPTION OF FINAL ARRANGEMENT *****
3847.      C         *****   OF W SEE REFERENCE 2, PP. 50-51.          *****
3848.      C
3849.      534       RETURN
3850.                END
3851.      C
3852.      C
3853.      C
3854.      C
3855.      C
3856.                SUBROUTINE SRT
3857.      C
3858.      C
3859.      C         **********************************************************************
3860.      C         *      REARRANGE THE COLUMNS ACCORDING TO ORDERS FROM SUBROUTINE      *
3861.      C         *      PART15.  A CALL TO THIS SUBROUTINE CANNOT RESULT IN A VIOLATION *
3862.      C         *      OF A PREVIOUS SORTING.                                         *
3863.      C         **********************************************************************
3864.      C
3865.      C
3866.                 DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
3867.               1BB(61,61),C(51),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
3868.               2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
3869.               3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
3870.               4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
3871.               6XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
3872.                COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
3873.               XINDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
3874.               XLPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,P,SUM,SUM1,SZERO,U,UEP,
3875.               XNOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
3876.               XULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
3877.               XXPN2M1,V,VV,ZB,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
3878.                MM=N-N3
3879.      C
3880.      C         *****   STORE THE ROW THAT DETERMINES THE SORING IN III.   *****
3881.      C
3882.                III=IPERM(LL)
3883.                DO 10 J=1,MM
3884.                  KPERM(J)=JPERM(J)
3885.      10        CONTINUE
3886.                MMM=0
3887.                DO 11 K=1,NN
3888.                  KK=INDEXQ(K+1)-1
3889.                  JJ=0
3890.                  MMM=MMM+1
3891.                  INDEXT(MMM)=INDEXQ(K)
3892.                  ML=INDEXQ(K+1)-INDEXQ(K)
3893.                  IF(SORT.EQ.2.0)GO TO 20
3894.      C
3895.      C         *****   SORT=1.0 PUTS THE NONNEGATIVE ENTRIES FIRST.   *****
3896.      C
3897.                  LL=INDEXQ(K)
3898.                  DO 12 J=LL,KK
3899.                    JJJ=JPERM(J)
```

```
3900.                IF(V(III,JJJ).GE.0.0)JJ=JJ+1
3901.       12       CONTINUE
3902.             GO TO 21
3903.       C
3904.       C     *****   SORT=2.0 PUTS THE NONPOSITIVE ENTRIES FIRST.   *****
3905.       C
3906.       20       LL=INDEXQ(K)
3907.                DO 22 J=LL,KK
3908.                   JJJ=JPERM(J)
3909.                   IF(V(III,JJJ).LE.0.0)JJ=JJ+1
3910.       22       CONTINUE
3911.       21       CONTINUE
3912.                IF(JJ.EQ.0.OR.JJ.EQ.ML)GO TO 11
3913.                MMM=MMM+1
3914.                INDEXT(MMM)=INDEXQ(K)+JJ
3915.                KKK=INDEXQ(K)-1
3916.                LLL=JJ+KKK
3917.                IF(SORT.EQ.2.0)GO TO 25
3918.       C
3919.       C     *****   SORT=1.0 PUTS THE NONNEGATIVE ENTRIES FIRST.   *****
3920.       C
3921.                LL=INDEXQ(K)
3922.                DO 14 J=LL,KK
3923.                   JJJ=KPERM(J)
3924.                   IF(V(III,JJJ).GE.0.0)GO TO 15
3925.                   LLL=LLL+1
3926.                   JPERM(LLL)=KPERM(J)
3927.                   GO TO 14
3928.       15          KKK=KKK+1
3929.                   JPERM(KKK)=KPERM(J)
3930.       14          CONTINUE
3931.                GO TO 26
3932.       C
3933.       C     *****   SORT=2.0 THE PUTS NONPOSITIVE ENTRIES FIRST.   *****
3934.       C
3935.       25       LL=INDEXQ(K)
3936.                DO 27 J=LL,KK
3937.                   JJJ=KPERM(J)
3938.                   IF(V(III,JJJ).LE.0.0)GO TO 28
3939.                   LLL=LLL+1
3940.                   JPERM(LLL)=KPERM(J)
3941.                   GO TO 27
3942.       28          KKK=KKK+1
3943.                   JPERM(KKK)=KPERM(J)
3944.       27          CONTINUE
3945.       26       CONTINUE
3946.       11       CONTINUE
3947.                NN=MMM
3948.                INDEXQ(NN+1)=N-N3+1
3949.                DO 16 J=1,NN
3950.                   INDEXQ(J)=INDEXT(J)
3951.       16          CONTINUE
3952.       13       RETURN
3953.                END
3954.       C
3955.       C
3956.       C
3957.       C
3958.       C
3959.                SUBROUTINE PART16
```

```
C
C
C     ***************************************************************
C     *      IF ANY EXIST, OBTAIN ALL THE FEASIBLE COMPLETIONS FOR THE GROUP     *
C     *      4 VARIABLES, GIVEN X(1),...,X(N3) AND ZB.                *
C     ***************************************************************
C
C
      DIMENSION A(61,61),ALMAX(61),AL(61),ALEP(61),ALLP(61),B(61),
     1BB(61,61),C(61),DELTA(61),D1(61,61),D2(61,61),INDEXQ(61),
     2INDEXT(61),IPERM(61),JPERM(61),JPM(61),KPERM(61),LPERM(61),
     3RHO(61),RHOP(61),TEMP(61),THOLD(61),TRI(61),U(61),UU(61),
     4UEP(61),ULP(61),UMIN(61),V(61,61),X(61),VV(61),XB(61),XX(61,61),
     5XPRIME(61),Y(61),JB(61),XF(61),KN(61),Y1(61),Y2(61)
      DIMENSION DROP(61),TOP(61)
      COMMON A,AAA,AL,ALEP,ALLP,ALMAX,AMAX,AMIN,B,BB,C,DELTA,DN2P,I,II,
     1INDEXQ,INDEXT,INDIC,IPERM,J,JJ,JPERM,JPM,K,KK,KL,KPERM,L,LK,LL,
     2LPERM,LZ,M,MAX,MIN,N,NB,N1,N2,N3,NN,NQ,R,SUM,SUM1,SZERO,U,UEP,
     3NOSOLN,NZERP,RHO,RHOP,RHP,RN2,RN2M1,SK,SK1,SOLN,SONE,SORT,UBAR,
     4ULP,UMIN,UU,UT,T,TEMP,THETA,TRI,TT,VBAR,VT,W,X,XB,XPRIME,XX,XPN2,
     5XPN2M1,V,VV,ZP,ZF,ZX,JB,Y,XF,KN,D1,D2,Y1,Y2,THOLD
  300 FORMAT('-','THE NEW FEASIBLE COMPLETION IS:')
  301 FORMAT('0','THE OBJECTIVE FUNCTION VALUE IS',T65,F8.2,'.')
  304 FORMAT(1H ,15F8.2)
      MM=N-N3
      MMM=NB+1
      NPLUS1=N+1
  200 CONTINUE
C
C     *****   STEP A(REFERENCE 2, P. 52).  RECALL LPERM PUTS THE          *****
C     *****   VARIABLES BACK INTO JB ORDER, JPM ORDERS THE COLUMNS OF      *****
C     *****   W, IPERM ORDERS THE ROWS OF W.  FIND THE CURRENT UPPER       *****
C     *****   AND LOWER BOUNDS FOR THE GROUP 4 VARIABLES, STORE THEM       *****
C     *****   IN UMIN AND ALMAX RESPECTIVELY                               *****
C
      MM=N-N3
      DO 10 J=1,MM
        LL=JPM(J)
        JJJ=LPERM(LL)
        IF(UEP(JJJ).GT.ULP(JJJ))GO TO 11
        UMIN(JJJ)=UEP(JJJ)
        GO TO 12
   11   UMIN(JJJ)=ULP(JJJ)
   12   IF(ALEP(JJJ).LT.ALLP(JJJ))GO TO 13
        ALMAX(JJJ)=ALEP(JJJ)
        GO TO 10
   13   ALMAX(JJJ)=ALLP(JJJ)
   10   CONTINUE
C
C     *****   STEP B(REFERENCE 2, P. 52).    *****
C
      MM=N-N3
      DO 540 J=1,MM
        LL=JPM(J)
        JJJ=LPERM(LL)
        TRI(J)=UMIN(JJJ)-ALMAX(JJJ)
        IF(TRI(J).LT.0.0) GO TO 541
  540   CONTINUE
C
C     *****   STEP C(REFERENCE 2, PP. 52-53).    *****
```

```
      C
            DO 542 J=1,MM
              LL=JPM(J)
              JJJ=LPERM(LL)
              X(JJJ)=ALMAX(JJJ)
  542       CONTINUE
            MMM=NB+1
            NPLUS1=N+1
            V(NPLUS1,NPLUS1)=-ZB/(ZX-ZF)
            DO 543 I=1,MMM
              III=IPERM(I)
              T=V(III,NPLUS1)
              DO 544 J=1,NB
                JJJ=JPERM(J)
                T=T+V(III,JJJ)*X(J)
  544         CONTINUE
              DO 545 J=1,MM
                JJJ=JPM(J)
                LL=LPERM(JJJ)
                T=T+V(III,JJJ)*X(LL)
  545         CONTINUE
              RHOP(III)=T
  543       CONTINUE
            K=IPERM(1)
            RHO(K)=RHOP(K)
      C
      C     *****   STEP D(REFERENCE 2, P. 53).   *****
      C
  546       I=1
            J=1
            F=1.0
      C
      C     *****   STEP E(REFERENCE 2, P. 53).   *****
      C
  547       K=IPERM(I)
            IF(RHO(K).GE.0.0)GO TO 548
            F=0.0
      C
      C     *****   STEP F(REFERENCE 2, P. 53).   *****
      C
  549       K=IPERM(I)
            L=JPM(J)
            IF(V(K,L).NE.0.0.AND.TRI(J).NE.0.0)GO TO 551
      C
      C     *****   STEP G(REFERENCE 2, P. 53).   *****
      C
  550       IF(J.EQ.MM)GO TO 552
            J=J+1
            GO TO 549
      C
      C     *****   STEP H(REFERENCE 2, P. 53).   *****
      C
  551       K=IPERM(I)
            L=JPM(J)
            IF(V(K,L))554,550,553
      C
      C     *****   STEP I(REFERENCE 2, P. 53).   *****
      C
  554       L=JPM(J)
            LL=LPERM(L)
```

```
4080.                IF(X(LL).LE.ALMAX(LL))GO TO 550
4081.          C
4082.          C     *****   STEP J(REFERENCE 2, P. 53).    *****
4083.          C
4084.          555   CONTINUE
4085.                IF(J.GE.MM)GO TO 552
4086.          C
4087.          C     *****   STEP K(REFERENCE 2, P. 54).    *****
4088.          C
4089.                K=IPERM(I)
4090.                L=JPM(J)
4091.                LL=LPERM(L)
4092.                RHO(K)=RHO(K)-V(K,L)*(X(LL)-ALMAX(LL))
4093.                X(LL)=ALMAX(LL)
4094.                J=J+1
4095.          C
4096.          C     *****   STEP L(REFERENCE 2, P. 54).    *****
4097.          C
4098.                K=IPERM(I)
4099.                IF(RHO(K))549,556,556
4100.          C
4101.          C     *****   STEP M(REFERENCE 2, P. 54).    *****
4102.          C
4103.          556   LL=JPM(J)
4104.                L=LPERM(LL)
4105.                IF(X(L).EQ.UMIN(L))GO TO 557
4106.                X(L)=X(L)+1.0
4107.                K=IPERM(I)
4108.                RHO(K)=RHO(K)+V(K,LL)
4109.          C
4110.          C     *****   STEP N(REFERENCE 2, P. 54).    *****
4111.          C
4112.                IF(RHO(K).LT.0.0)GO TO 555
4113.                LLL=J-1
4114.                IF(LLL.EQ.0) GO TO 548
4115.                DO 558 LL=1,LLL
4116.                  LML=JPM(LL)
4117.                  K=LPERM(LML)
4118.                  X(K)=ALMAX(K)
4119.          558     CONTINUE
4120.                GO TO 548
4121.          C
4122.          C     *****   STEP O(REFERENCE 2, P. 54).    *****
4123.          C
4124.          557   IF(J.EQ.MM)GO TO 552
4125.          C
4126.          C     *****   STEP P(REFERENCE 2, P. 54).    *****
4127.          C
4128.                K=IPERM(I)
4129.                L=JPM(J)
4130.                IF(V(K,L).LT.0.0)GO TO 559
4131.                J=J+1
4132.                GO TO 556
4133.          559   LL=LPERM(L)
4134.                X(LL)=ALMAX(LL)
4135.                RHO(K)=RHO(K)-V(K,L)*TRI(J)
4136.                J=J+1
4137.                GO TO 556
4138.          C
4139.          C     *****   STEP Q(REFERENCE 2, PP. 54-55).    *****
```

```
C
553   L=JPM(J)
      LL=LPERM(L)
      IF(X(LL).GE.UMIN(LL))GO TO 550
      K=IPERM(I)
      RHO(K)=RHO(K)+V(K,L)*(UMIN(LL)-X(LL))
C
C     *****   STEP R(REFERENCE 2, P. 55).   *****
C
      IF(RHO(K).GE.0.0)GO TO 560
      LL=LPERM(L)
      X(LL)=UMIN(LL)
      GO TO 550
560   LL=LPERM(L)
      X(LL)=UMIN(LL)-1.0
      RHO(K)=RHO(K)-V(K,L)
C
C     *****   STEP S(REFERENCE 2, P. 55).   *****
C
561   IF(RHO(K).LT.0.0)GO TO 562
C
C     *****   STEP T(REFERENCE 2, P. 55).   *****
C
      LL=LPERM(L)
      X(LL)=X(LL)-1.0
      RHO(K)=RHO(K)-V(K,L)
      GO TO 561
562   LL=LPERM(L)
      X(LL)=X(LL)+1.0
      RHO(K)=RHO(K)+V(K,L)
      KK=J-1
      IF(KK.LE.0) GO TO 548
      DO 563 LL=1,KK
        LML=JPM(LL)
        K=LPERM(LML)
        X(K)=ALMAX(K)
563     CONTINUE
C
C     *****   STEP U(REFERENCE 2, P. 55).   *****
C
548   TOL=10.0**(-4)
      IF(I.GT.NR)GO TO 564
      I=I+1
      K=IPERM(I)
      T=RHOP(K)
      DO 565 LL=J,MM
        KK=JPM(LL)
        LML=LPERM(KK)
        T=T+V(K,KK)*(X(LML)-ALMAX(LML))
565     CONTINUE
      IF(ABS(T).LE.TOL)T=0.0
      RHO(K)=T
      J=1
      GO TO 547
C
C     *****   STEP V(REFERENCE 2, P. 55).   *****
C
564   CONTINUE
      IF(F.NE.0.0)GO TO 566
      K=IPERM(1)
```

```
      T=RHOP(K)
      DO 567 LL=J,MM
        L=JPM(LL)
        LML=LPERM(L)
        T=T+V(K,L)*(X(LML)-ALMAX(LML))
  567   CONTINUE
      RHO(K)=T
      GO TO 546
C
C     *****   STEP W(REFERENCE 2, PP. 55-56).   *****
C
  566 IF(MMM.GT.M)GO TO 571
      DO 568 I=MMM,M
        T=0.0
        DO 569 J=1,N
          JJ=JPERM(J)
          T=T+A(I,JJ)*X(J)
  569     CONTINUE
        T=T-B(I)
        IF(T.GT.0.0)GO TO 570
  568   CONTINUE
      GO TO 571
  570 J=1
C
C     *****   STEP X(REFERENCE 2, P. 56).   *****
C
  572 K=JPM(J)
      L=LPERM(K)
      IF(X(L).EQ.UMIN(L)) GO TO 573
      X(L)=X(L)+1.0
      K=IPERM(1)
      L=JPM(J)
      RHO(K)=RHO(K)+V(K,L)
      GO TO 546
C
C     *****   STEP Y(REFERENCE 2, P. 56).   *****
C
  573 IF(J.EQ.MM)GO TO 552
      K=IPERM(1)
      L=JPM(J)
      LL=LPERM(L)
      RHO(K)=RHO(K)-V(K,L)*(X(LL)-ALMAX(LL))
      X(LL)=ALMAX(LL)
      J=J+1
      GO TO 572
C
C     *****   STEP Z(REFERENCE 2, P. 56).   *****
C
  571 T=0.0
      DO 574 J=1,N
        LL=JPERM(J)
        T=T+C(LL)*X(J)
  574   CONTINUE
      IF(T.GT.ZB)GO TO 575
      J=1
      GO TO 572
C
C     *****   A NEW FEASIBLE COMPLETION HAS BEEN FOUND.   *****
C     *****   RESET ZB AND XB.  NOTE THAT 'FEASIBLE'      *****
C     *****   IMPLIES THAT XB HAS A BETTER OBJECTIVE      *****
```

```
C      *****   FUNCTION VALUE THAN THAT OF THE CURRENT      *****
C      *****   COMPLETION.                                  *****
C
575   CONTINUE
      ZB=T
      DO 576 J=1,N
        XB(J)=X(J)
576     CONTINUE
      DO 305 J=1,N
        TOP(J)=XB(J)
305     CONTINUE
C
C      *****   PUT XB IN THE ORIGINAL JB ORDER AND STORE IT IN TOP.   *****
C
      CALL PERMUT(TOP,N,LPERM)
      DO 302 J=1,N
        LL=JB(J)
        DROP(LL)=TOP(J)
302     CONTINUE
      WRITE(6,300)
      WRITE(6,304)(DROP(J),J=1,N)
      WRITE(6,301)ZB
      J=1
      GO TO 572
552   CONTINUE
541   SOLN=0.0
      RETURN
      END
```

APPENDIX II

LISTING OF SAMPLE INPUT FOR COMPUTER PROGRAM BDSCAN

```
//GO.SYSIN DD *
TEST PROBLEM
   15   16    1
  22.  40.  33. -36.  -1.  43.  56.   8.   6. -38.  12.   9. -16.  47.  33.
   5.
  19.  10.  49.  13.  26.  28.  38.  50.  17.  36.  29. -37.  23. -33.  46.
   5.
  17.  53. -16.  34. -30.  47.  13. -39. -25. -17.  42.  27.  50.  39.   3.
   5.
   4. -36. -31. -21.  34.  35. -16.  37.  11. -12. -15.  35. -32. -11. -33.
   5.
  16.   4. -23.  15.  10. -40.  27.  25. -14. -21. -40. -39.  56. -16.  30.
   5.
  -2. -22. -26.  22.  44. -33. -10.  58. -11.  47.  55. -24.   5. -30.  28.
   5.
 -33.   8.  16. -12.  27.  15.  16. -23.  26.  31.  10.  22.  35. -30. -31.
   5.
 -27.  30.  53.   5. -36. -14.   0.   0.  39.  18.  38. -36.  59.  51. -33.
   5.
  28.  12. -12. -25. -24. -34.  58. -27.  38. -30.  11.  29.   2.  40.  51.
   5.
 -38.  25. -40.  50.  -3.   9.   5. -36.  21.  28.  29. -28.  56.  16.  35.
   5.
 -24. -17.   0.  49. -35.  24.  34.  33. -11.  34.  42.  49.  56.   4.   3.
   5.
   8. -18.  13.  -5.  36.  20.  46.  36.  53. -16.  14.  29.  31.   5.  -9.
   5.
  58.  21.  38.  -6.  35.  -7.  41.  34.  48.   8. -37. -31.  15.  48.  55.
   5.
 -32. -20.  27.  27.  25. -28.  16.  -8.  43.  59. -11.   4.  14.   3.  42.
   5.
  26.  17. -32. -21.  47. -37. -15.  20. -22.  10.   5.  34.  49. -27. -29.
   5.
 834. 853. 979. 822. 812. 752. 822. 588. 911. 915. 979. 584. 810. 720. 590.
  58.   6.  60.  63.  -6.  71.  56. -14.  53.  43. -10.   3. -10.  -4.  52.
  60.
      0.0          0.0          6.33003      0.0          0.0         -3.82294
      0.0         -0.09148      0.0         -8.41045      0.0         -2.24632
    -1.60377       9.19545      0.0          0.0          0.0         -2.06989
      0.0          0.0         -1.40126      0.0         -0.76006      0.0
     2.51634       0.0          0.45438      1.70634     -0.45927      0.0
      0.0          0.0          0.43273      0.0          0.0          5.94474
      0.0         -1.15333      0.0          3.29456      0.0         -0.36586
    -1.41647      -5.29791      0.0          0.0          0.0         -1.45657
      1.0          0.0          1.32734      0.0          0.02374      0.0
     2.25161       0.0          1.05993      0.62613     -2.39565      0.0
      0.0          0.0          3.83045      0.0          1.0         -3.42990
      0.0         -0.39525      0.0         -5.04976      0.0         -1.02417
    -1.41106       5.78087      0.0          0.0          1.0         -2.22353
```

```
   0.0              0.0              2.70460          0.0             -0.38344          0.0
  -0.23687          0.0             -0.80698         -1.43725         -1.08842          0.0
   0.0              0.0             -1.68813          0.0              0.0             -0.20241
   0.0              0.46843          0.0              1.20498          1.0             -0.02652
   1.05218         -1.52227          0.0              0.0              0.0              5.39466
   0.0              0.0             -5.14769          0.0             -1.21517          0.0
   6.09729          0.0             -0.31908         -1.46839          7.85910          0.0
   0.0              0.0              2.40270          0.0              0.0              2.04752
   0.0             -0.70377          0.0             -2.63566          0.0             -0.87693
   0.25549          0.23492          0.0              0.0              0.0             -8.80472
   0.0              0.0              3.51965          0.0              7.93484          0.0
  10.54063          0.0             12.68879          6.61352         -8.15562          0.0
   0.0              0.0              0.41573          0.0              0.0             -1.03597
   0.0             -1.21444          0.0              0.10630          0.0             -0.04526
   0.56556          0.96335          1.0
10  15   4  20   21  18  23   2  25  17  27   6   1  16  31
   2.81693          1.25460          0.0              6.63173          0.0              0.48790
   0.0              0.0              0.0              0.31753          0.0              0.0
   0.0              0.0              0.75636        123.86716
   1.0              1.0              0.0              8.0              0.0              0.0
   0.0              0.0              0.0              0.0              0.0              0.0
   0.0              0.0              0.0            119.0
```

APPENDIX III

LISTING OF SAMPLE OUTPUT FOR COMPUTER PROGRAM BDS(

```
      TEST PROBLEM
THE CONSTRAINT MATRIX A IS 15  BY 16.
THE CONSTRAINT MATRIX A(I,J) IS:
   22.00    40.00    33.00   -36.00    -1.00    43.00    56.00     8.00     6.00   -38.00    12.00     9.00   -16.00    47.00    33.00
    5.00
   19.00    10.00    49.00    13.00    26.00    28.00    38.00    50.00    17.00    36.00    29.00   -37.00    23.00   -33.00    46.00
    5.00
   17.00    53.00   -16.00    34.00   -30.00    47.00    13.00   -39.00   -25.00   -17.00    42.00    27.00    50.00    39.00     3.00
    5.00
    4.00   -36.00   -31.00   -21.00    34.00    35.00   -16.00    37.00    11.00   -12.00   -15.00    35.00   -32.00   -11.00   -33.00
    5.00
   16.00     4.00   -23.00    15.00    10.00   -40.00    27.00    25.00   -14.00   -21.00   -40.00   -39.00    56.00   -16.00    30.00
    5.00
   -2.00   -22.00   -26.00    22.00    44.00   -33.00   -10.00    58.00   -11.00    47.00    55.00   -24.00     5.00   -30.00    28.00
    5.00
  -33.00     8.00    16.00   -12.00    27.00    15.00    16.00   -23.00    26.00    31.00    10.00    22.00    35.00   -30.00   -31.00
    5.00
  -27.00    30.00    53.00     5.00   -36.00   -14.00     0.0      0.0     39.00    18.00    38.00   -36.00    59.00    51.00   -33.00
    5.00
  -28.00    12.00   -12.00   -25.00   -24.00   -34.00    58.00   -27.00    38.00   -30.00    11.00    29.00     2.00    40.00    51.00
    5.00
  -38.00    25.00   -40.00    50.00    -3.00     9.00     5.00   -36.00    21.00    28.00    29.00   -28.00    56.00    16.00    35.00
    5.00
  -24.00   -17.00     0.0     49.00   -35.00    24.00    34.00    33.00   -11.00    34.00    42.00    49.00    56.00     4.00     3.00
    5.00
    8.00   -18.00    13.00    -5.00    36.00    20.00    46.00    36.00    53.00   -16.00    14.00    29.00    31.00     5.00    -9.00
    5.00
   58.00    21.00    38.00    -6.00    35.00    -7.00    41.00    34.00    48.00     8.00   -37.00   -31.00    15.00    48.00    55.00
    5.00
  -32.00   -20.00    27.00    27.00    25.00   -28.00    16.00    -8.00    43.00    59.00   -11.00     4.00    14.00     3.00    42.00
    5.00
   26.00    17.00   -32.00   -21.00    47.00   -37.00   -15.00    20.00   -22.00    10.00     5.00    34.00    49.00   -27.00   -29.00
    5.00
THE RIGHT HAND SIDE B(I) IS:
  834.00   853.00   979.00   822.00   812.00   752.00   822.00   588.00   911.00   915.00   979.00   584.00   810.00   720.00   590.00
THE COST COEFFICIENTS C(J) ARE:
   58.00     6.00    63.00    63.00    -6.00    71.00    56.00   -14.00    53.00    43.00   -10.00     3.00   -10.00    -4.00    52.00
   60.00
FOLLOWING, IN ORDER OF INCREASING I, IS THE ITH BASIC VARIABLE, INCLUDING SLACKS:
   10   15    4   20   21   18   23    2   25   17   27    6    1   16   31
THE LP SOLUTION IS:
    2.82     1.25     0.0      6.63     0.0      0.49     0.0      0.0      0.0      0.32     0.0      0.0      0.0      0.0      0.76
  123.87
THE SUBOPTIMAL FEASIBLE SOLUTION IS:
    3.00     1.00     0.0      8.00     0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
  119.00
THE NUMBER OF VARIABLES BASIC IN THE LP SOLUTION IS  7.
THE FIRST N EXTREME POINTS OF THE N-SIMPLEX FORMED BY THE N
CONSTRAINTS THAT ARE BINDING ON THE LP SOLUTION PLUS THE
CONSTRAINT CX = GF = ZE ARE:
   -2.35     0.94     0.0      7.56     0.0    -11.11     0.0      0.0      0.0    -13.29     0.0      0.0      0.0      0.0      5.21
  142.80
   -3.42     5.47     0.0    -11.46     0.0     16.16     0.0      0.0      0.0     11.95     0.0      0.0      0.0      0.0      5.02
  113.15
    3.66    -0.34     0.0      8.06     0.0      1.94     0.0      0.0      0.0      0.43     0.0      0.0      0.0      0.0      1.66
  114.40
   31.75     1.99     0.0    -29.54     0.0     67.31     0.0      0.0      0.0     92.65     0.0      0.0      0.0      0.0    -26.87
    8.15
    3.24     1.81     0.0      6.81     0.0      0.64     0.0      0.0      0.0      1.41     0.0      0.0      0.0      0.0      0.54
  117.70
    3.16    -0.15     0.0      8.51     0.0      2.43     0.0      0.0      0.0      2.44     0.0      0.0      0.0      0.0     -1.50
  115.10
    2.26     1.02     0.0     19.12     0.0    -18.04     0.0      0.0      0.0    -21.36     0.0      0.0      0.0      0.0      1.84
  143.09
    3.31    -0.38     1.98    15.00     0.0     -5.88     0.0      0.0      0.0     -7.68     0.0      0.0      0.0      0.0      1.08
  121.14
    2.69     2.51     0.0      5.70     1.26     2.08     0.0      0.0      0.0      2.40     0.0      0.0      0.0      0.0     -0.01
  117.52
    3.27     1.30     0.0      8.33     0.0     -0.77     0.98     0.0      0.0     -0.15     0.0      0.0      0.0      0.0      0.13
  118.35
    2.13     2.29     0.0      4.60     0.0      4.12     0.0      1.12     0.0      3.71     0.0      0.0      0.0      0.0      0.93
  115.21
    4.04     1.01     0.0      8.71     0.0      0.19     0.0      0.0      0.70     0.63     0.0      0.0      0.0      0.0     -0.20
  116.13
    1.57     2.14     0.0      0.63     0.0      4.86     0.0      0.0      0.0      4.21     1.60     0.0      0.0      0.0      2.76
  117.13
    0.86     4.72     0.0     14.94     0.0    -21.60     0.0      0.0      0.0    -20.30     0.0      5.08     0.0      0.0      4.82
  150.74
    3.37     0.80     0.0      6.55     0.0      0.08     0.0      0.0      0.0      1.22     0.0      0.0      0.72     0.0      0.60
  117.76
    3.46    -0.33     0.0      9.49     0.0     -1.17     0.0      0.0      0.0     -1.33     0.0      0.0      0.0      1.29     0.32
  119.27
```

```
THE WEIGHTS ON THESE N EXTREME POINTS ARE BOUNDED ABOVE BY:
    0.87     0.63     0.94     0.16     1.00     0.77     0.81     0.92     1.00     1.00     0.90     0.96     0.62     0.82     0.0
    0.99
THE UNCONDITIONAL BOUNDS ON THE VARIABLES ARE:
 LOWER BOUNDS:
    0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
   97.00
 UPPER BOUNDS:
    8.00     5.00     1.00    18.00     1.00    21.00     0.0      1.00     0.0     23.00     1.00     4.00     0.0      1.00     5.00
  149.00
THE PRODUCT OF THE UNCONDITIONAL RANGES IS                              361670400.
THE NEW FEASIBLE COMPLETION IS:
    3.00     1.00     0.0      7.00     0.0      1.00     0.0      0.0      0.0      1.00     0.0      0.0      0.0      0.0      0.0
  119.00
THE OBJECTIVE FUNCTION VALUE IS                                         7875.00.
THE NEW FEASIBLE COMPLETION IS:
    3.00     1.00     0.0      7.00     0.0      1.00     0.0      0.0      0.0      1.00     0.0      0.0      0.0      0.0      0.0
  120.00
THE OBJECTIVE FUNCTION VALUE IS                                         7935.00.
THE ELIGIBLE GROUP 1 SOLUTIONS HAVE BEEN SCANNED THROUGH VARIABLE    7.
AT THE END OF THE ALGORITHM THE FOLLOWING INFORMATION IS KNOWN:
 THE FEASIBILITY TEST SLACKS ARE:
    7.00    12.00     0.0     17.00     9.00    56.00    16.00   375.00    31.00   370.00   116.00   351.00   622.00    67.00    69.00
 THE NUMBER OF TIMES THROUGH PART 1 WAS            1.
 THE NUMBER OF TIMES THROUGH PART 2 WAS            1.
 THE NUMBER OF TIMES THROUGH PART 3 WAS            1.
 THE NUMBER OF TIMES THROUGH PART 4 WAS            1.
 THE NUMBER OF TIMES THROUGH PART 5 WAS            7.
 THE NUMBER OF TIMES THROUGH PART 6 WAS            1.
 THE NUMBER OF TIMES THROUGH PART 7 WAS            5.
 THE NUMBER OF TIMES THROUGH PART 8 WAS            5.
 THE NUMBER OF TIMES THROUGH PART 9 WAS           24.
 THE NUMBER OF TIMES THROUGH PART10 WAS           24.
 THE NUMBER OF TIMES THROUGH PART11 WAS            1.
 THE NUMBER OF TIMES THROUGH PART12 WAS          214.
 THE NUMBER OF TIMES THROUGH PART13 WAS           21.
 THE NUMBER OF TIMES THROUGH PART14 WAS          214.

 THE NUMBER OF TIMES THROUGH PART15 WAS            1.
 THE NUMBER OF TIMES THROUGH PART16 WAS          214.
 THE OBJECTIVE FUNCTION VALUE FOR THE SUBOPTIMAL SOLUTION IS     7824.00.
 THE OBJECTIVE FUNCTION VALUE FOR THE LP SOLUTION IS             8108.36.
 THE NORMALIZED DIFFERENCE BETWEEN THE OPTIMAL LP AND ILP OBJECTIVE FUNCTION VALUES IS          0.99160.
 THE NORMALIZED DIFFERENCE BETWEEN THE OPTIMAL ILP AND THE SUBOPTIMAL OBJECTIVE FUNCTION VALU  0.63491.
 THE FINAL INTEGER SOLUTION IS:
    3.00     1.00     0.0      7.00     0.0      1.00     0.0      0.0      0.0      1.00     0.0      0.0      0.0      0.0      0.0
  120.00
 THE OBJECTIVE FUNCTION VALUE IS                 7935.00.
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: No. 87	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): DOCUMENTATION OF A COMPUTER PROGRAM FOR THE BOUND-AND-SCAN ALGORITHM FOR INTEGER LINEAR PROGRAMMING,	5. TYPE OF REPORT & PERIOD COVERED: TECHNICAL REPORT, / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): NANCY E. JACQMIN	8. CONTRACT OR GRANT NUMBER(s): N00014-76-C-0418, NSF-MCS 76-81259
9. PERFORMING ORGANIZATION NAME AND ADDRESS: DEPT. OF OPERATIONS RESEARCH, STANFORD UNIVERSITY, STANFORD, CALIF.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-061
11. CONTROLLING OFFICE NAME AND ADDRESS: OPERATIONS RESEARCH PROGRAM, OFFICE OF NAVAL RESEARCH CODE 434, ARLINGTON, VA. 22217	12. REPORT DATE: JULY 1979 / 13. NUMBER OF PAGES: 87
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): APPROVED FOR PUBLIC RELEASE AND SALE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES: This research was supported in part by National Science Foundation Grant NSF Number - MCS 76-81259 A01, Dept. of Operations Research, Stanford University.	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): INTEGER PROGRAMMING, ALGORITHMS, IMPLICIT ENUMERATION, SOFTWARE	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number): This report gives documentation for the computer code BDSCAN for solving integer linear programming problems using the bound-and-scan algorithm. A listing of the program with sample input and output is included.	

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

402 766

```
                T=T+V(K,L)*(X(LML)-ALMAX(LML))
  567           CONTINUE
                PHO(K)=T
                GO TO 565
C
C               ****    STEP W(REFERENCE 2, P
C
                IF(MMM.GT.M)GO TO 571
                DO 568 I=MMM,M
                   T=0.0
                   DO 569 J=1,N
                      JJ=JPERM(J)
                      T=T+A(I,JJ)*X(J)
  569              CONTINUE
                   T=T-R(I)
                   IF(T.GT.0.0)GO TO 570
  568           CONTINUE
                GO TO 571
  570           J=1
C
C               ****    STEP X(REFERENCE 2, P
C
                K=JPM(J)
                L=LPERM(K)
                IF(X(L).EQ.UMIN(L)) GO TO 573
                X(L)=X(L)+1.0
                K=IPERM(L)
                L=JPM(J)
                PHO(K)=RHO(K)+V(K,L)
                GO TO 546
C
C               ****    STEP Y(REFERENCE 2, P
C
  573           IF(J.EQ.MM)GO TO 552
                K=IPERM(L)
                L=JPM(J)
                LL=LPERM(L)
                PHO(K)=RHO(K)-V(K,L)*(X(LL)-AL
                X(LL)=ALMAX(LL)
                J=J+1
                GO TO 572
C
C               ****    STEP Z(REFERENCE 2, P
C
  571           T=0.0
                DO 574 J=1,N
                   LL=JPERM(J)
                   T=T+C(LL)*X(J)
  574           CONTINUE
                IF(T.GT.ZB)GO TO 575
                J=1
                GO TO 572
C
C               ****    A NEW FEASIBLE SOLUTIO
C               ****    RESET ZB AND XB.  NOTE
C               ****    IMPLIES THAT XB HAS A
```