LAMPIRAN A. LISTING PROGRAM

INTEGER AAAA, NNNN
PARAMETER (AAAA=12400, NNNN=1230)

INTEGER LNODE, MNODE
INTEGER DOWN(NNNN), NEXT(NNNN), LEVEL(NNNN)
INTEGER ARCID(NNNN), FLOW(NNNN), DUAL(NNNN), CAT(NNNN)
EQUIVALENCE (DUAL(1), CAT(1))

C

NODE FORMAT

1 - PREDECESSOR OR DOWN POINTER
2 - THREAD OR NEXT POINTER
3 - LEVEL NUMBER
4 - ASSOCIATED ARC IDENTIFIER
   (+=ORIENTATION OPPOSITE TO DOWN POINTER)
   (=-ORIENTATION SAME AS DOWN POINTER)
5 - FLOW ON ARC
6 - DUAL VARIABLE VALUE

INTEGER LARC, MARC
INTEGER FROM(AAAA), COST(AAAA), CAPAC(AAAA)
INTEGER FLOOR(AAAA), NAME(AAAA)
INTEGER ARCNAM

ARC FORMAT

1 - FROM NODE
2 - COST
3 - CAPACITY(-, IF AT UB)
4 - LOWER BOUND
5 - NAME

ARC(ARTY) IS GENERAL REFERENCE FOR ARTIFICIAL ARCS

INTEGER TO
INTEGER SLACK, ARTIF, DUMMY, XCESS
INTEGER PRICE0, TOO
INTEGER TRY, PRICE
INTEGER NEWARC, NEWFR
INTEGER NEWFRM, NEWTO
INTEGER ARTY, BIG, ARTYP1
INTEGER THD
INTEGER DW(2), CH(2), DWN, CHG
INTEGER THETA, JTHETA, KTHETA, POSS, JPOSS
INTEGER FRM, LVJ
INTEGER FM, LST
INTEGER DWE
INTEGER FLW, AID
INTEGER Q1, Q2
INTEGER DIR, REF
INTEGER U1, U2, U3, U4
LOGICAL INFEAS
LOGICAL OPTIM
LOGICAL DMP
LOGICAL PPR
LOGICAL PRES

DATA ITDOPT/ 1 /
DATA ARTY/ 1 /
DATA BIG/ 10000000 /
DATA U1, U2, U3, U4/5, 5, 5, 5/

DATA SLACK/ 66666/
DATA ARTIF/ 99999/
DATA DUMMY/ 77777/
DATA XCESS/ 88888/

1000 FORMAT(8I10)
1001 FORMAT(1X, 8I10)
1002 FORMAT(2X, I8, 5I10)
1003 FORMAT(1X, I10, 5I10)
1004 FORMAT(F10.0)
1005 FORMAT(1X, F10.5)
2001 FORMAT(1X, I5, 1H,, 4I15, 1X, A10)
2002 FORMAT(1X, I5, 1H,, 6I15)
8910 FORMAT(8H0PROBLEM, I5)

OPEN(5, FILE='NETFLO.INP')
OPEN(6, FILE='NETFLO.OUT')
ARTYP1 = ARTY+1
LNODE = NNNN-2
LARC = AAAA-2
LARCP1 = LARC+1

C NSTOP - EXIT CONDITION
C MNODE - NUMBER OF NODES
C NET - TOTAL BALANCE IN NETWORK
C MSORC - NUMBER OF SOURCES
C MSINK - NUMBER OF SINKS
C MARC - NUMBER OF ARCS
C MTREE - NUMBER OF BRANCHES ON TREE (EXCLUDING ROOT)
C THD - POINTER MOVING ALONG THREAD
C TRY - VARIABLE ENCOUNTERED DURING SETUP OR PRICING
C PRICE - REDUCED COST FOR TRY
C NEWARC - BEST VARIABLE TO ENTER
C NEWPR - PRICE FOR NEWARC
C DW - DOWN PTRS FOR RATIO TEST (FROM STEM, TO STEM)
C CH - PATH CONDITIONS FOR RATIO TEST (FROM STEM, TO STEM)
C DWN - POINTER MOVING ALONG DOWN PATH
C CHG - PATH CONDITION
C THETA - MINIMUM RATIO IN RATIO TEST
C JTHETA - UB (-) OR LB (+1) CONDITION FOR MIN THETA
C KTHETA - MIN THETA OCCURS ON FROM STEM (1) OR TO STEM (2)
C POSS - CANDIDATE FOR MIN THETA
C JPOSS - UB (-1) OR LB (+1) CONDITION FOR CANDIDATE FOR MIN THETA
C DWE - ROOT OF CYCLE

OPEN(5, FILE='NETFLO.INP')
OPEN(6, FILE='NETFLO.OUT')
ARTYP1 = ARTY+1
LNODE = NNNN-2
LARC = AAAA-2
LARCP1 = LARC+1
C UNUSED NODE NUMBER
LNODP1 = LNODE+1
NPROB = 0
KARD = 0
5 CONTINUE
PPR = .TRUE.
PRES = .FALSE.
NPROB = NPROB+1
C INPUT NUMBER OF REGULAR NODES
NSTOP = 2
READ(U1,1000,END=999) I
C
C NUMBER OF NODES(I10)
C
KARD = KARD+1
IF(PPR) WRITE(6,8910) NPROB
WRITE(6,1001) I
NSTOP = 3
IF(I.LE.0) GO TO 999
MNODE = I
MNODP1 = MNODE+1
MNODP2 = MNODE+2
NSTOP = 4
IF(MNODP1.GT.LNODE) GO TO 999
DUAL(MNODP1) = 0
C INITIALIZE NODE ARRAY
DO 10 J10=1,MNODP1
DOWN(J10) = 0
NEXT(J10) = 0
LEVEL(J10) = 0
ARCID(J10) = 0
FLOW(J10) = 0
10 CONTINUE
C INITIALIZE ARTIFICIAL
FROM(ARTY) = MNODP1
COST(ARTY) = BIG
CAPAC(ARTY) = 0
FLOOR(ARTY) = 0
NAME(ARTY) = ARTIF
C
C DURING SETUP WE USE THE ABSENCE OF THE UPPER BOUND FLAG
C ON AN ARC TO INDICATE THAT THE ARC IS ELIGIBLE TO BECOME
C PART OF THE STARTING BASIS, OTHERWISE IT IS NOT.
C
NET = 0
MSORC = 0
MARC = 0
C INITIALIZE SUPPLY AND DEMAND LISTS WITH SELF-POINTER
NEXT(MNODP1) = MNODP1
DOWN(MNODP1) = MNODP1
C INPUT NON-ZERO NODAL BALANCES
15 CONTINUE
NSTOP = 5
READ(U2,1000,END=999) I,J
C
C NODE NUMBER(I10),BALANCE(I10)/BLANK= ESCAPE
C
KARD = KARD+1
WRITE(6,1001) I,J
NSTOP = 6
IF(I) 999,60,20
20 CONTINUE
   IF(I.GT.MNODE) GO TO 999
   NSTOP = 7
   IF(FLOW(I).NE.0) GO TO 999
   FLOW(I) = J
   NET = NET+J
   IF(J.LE.0) GO TO 15
   MSORC = MSORC+1
   SAVE ORIGINAL SUPPLY IN LEVEL
   LEVEL(I) = J
   NEXT(I) = NEXT(MNODP1)
   NEXT(MNODP1) = I
   GO TO 15
60 CONTINUE
   TEST FOR FEASIBILITY
   NSTOP = 8
   IF(NET.LT.0) GO TO 999
   INPUT NUMBER OF ARCS TO EACH NODE (IN NUMERIC ORDER)
   NSTOP = 9
   READ(U3,1000,END=999) (CAT(I),I=1,MNODE)
   NUMBER OF ARCS TO EACH NODE (8I10)
   KARD = KARD+1
   WRITE(6,1001) (CAT(I),I=1,MNODE)
   RESERVE LOCATIONS FOR INPUT ARCS BY FILLING WITH DUMMIES
   CAT(N) WILL POINT TO THE NEXT OPEN LOCATION FOR STORING
   ARCS WHOSE TERMINAL NODE IS N.
   NSTOP = 10
   I = 1
   J = ARTY
   DO 80 J80 = 1,MNODE
      I = -I
      K70 = MAX0(1,CAT(J80))
      IF(J+K70.GT.LARC) GO TO 999
      CAT(J80) = ISIGN(J+1,I)
      DO 70 I70=1,K70
         J = J+1
         FROM(J) = ISIGN(J80,I)
         COST(J) = 0
         CAPAC(J) = -BIG
         FLOOR(J) = 0
         NAME(J) = DUMMY
      70 CONTINUE
   80 CONTINUE
   MARC = J+1
   IF(MARC.GT.LARC) GO TO 999
   FROM(MARC) = ISIGN(MNODP1,-I)
   INPUT REGULAR ARCS
   KOST0 = 0
   90 CONTINUE
   NSTOP = 11
   READ(U4,1002,END=999) ARCNAM,I,J,K,L,M
   NAME, FROM NODE, TO NODE, COST, CAPACITY, LOWER BOUND
   (I10), (I10), (I10), (I10), (I10), (I10), (I10)
   /BLANK = ESCAPE
C
KARD = KARD+1
WRITE(6,1003)   ARCNAM,I,J,K,L,M
NSTOP = 12
IF(I) 999,150,100
100 CONTINUE
IF(I.GT.MNODE) GO TO 999
IF(J.GT.MNODE) GO TO 999
IF(J.LE.0) GO TO 999
NSTOP = 13
IF(L.GE.BIG) GO TO 999
IF(L.EQ.0) L = BIG
IF(L.LT.0) L = 0
IF(M.GE.BIG) GO TO 999
IF(M.LT.0) GO TO 999
IF(M.GT.L) GO TO 999
II = CAT(J)
JJ = IABS(II)
C     TEST TO SEE IF CATEGORY IS FULL
KK = ISIGN(LNODP1,II)
IF(IEOR(KK,FROM(JJ)).GT.0) GO TO 140
C     MOVE REST OF ARCS DOWN TO ACCOMODATE
NSTOP = 14
IF(MARC.EQ.LARC) GO TO 999
MARC = MARC+1
K120 = MARC-JJ
M120 = MARC
DO 120 J120=1,K120
L120 = M120-1
FROM(M120)  = FROM(L120)
COST(M120)  = COST(L120)
CAPAC(M120) = CAPAC(L120)
FLOOR(M120) = FLOOR(L120)
NAME(M120)  = NAME(L120)
M120 = L120
120 CONTINUE
DO 130 J130=J,MNODE
130 CAT(J130) = CAT(J130)+ISIGN(1,CAT(J130))
C     INSERT NEW ARC
140 CONTINUE
FROM(JJ) = ISIGN(I,II)
COST(JJ) = K
KOST0 = KOST0+K*M
CAPAC(JJ) = L-M
FLOOR(JJ) = M
FLOW(I) = FLOW(I)-M
FLOW(J) = FLOW(J)+M
NAME(JJ) = ARCNAM
CAT(J) = ISIGN(JJ+1,II)
C     MARK FROM NODE WITH ARC RUNNING FROM IT FLAG
ARCID(I) = -1
GO TO 90
150 CONTINUE
C
C     ELIMINATE NON-ESSENTIAL DUMMY ARCS
C     CAT(N) WILL POINT TO LOCATION OF FIRST ARC TERMINAL AT NODE N
C
I = LNODP1
K = ARTY
L = 0
MARC = MARC-1
DO 190 J190=ARTYP1,MARC
   J = FROM(J190)
   IF(IEOR(I,J).GT.0) GO TO 160
   I = -I
   L = L+1
   CAT(L) = K+1
   GO TO 170
160 IF(IABS(J).EQ.L) GO TO 190
170 K = K+1
    FROM(K) = FROM(J190)
    COST(K) = COST(J190)
    CAPAC(K) = CAPAC(J190)
    FLOOR(K) = FLOOR(J190)
    NAME(K) = NAME(J190)
190 CONTINUE
    MARC = K
    MREG = K
    NSTOP = 15
    IF(MARC+MAX0(1,MSORC)+1.GT.LARC) GO TO 999
C     ADD REGULAR SLACKS
C
   I = -FROM(MARC)
   THD = NEXT(MNODP1)
   NEXT(MNODP1) = MNODP1
   IF(THD.NE.MNODP1) GO TO 192
C     NO REGULAR SLACKS, ADD DUMMY
    MARC = MARC+1
    FROM(MARC) = ISIGN(MNODP1,I)
    COST(MARC) = 0
    CAPAC(MARC) = -BIG
    FLOOR(MARC) = 0
    NAME(MARC) = DUMMY
    GO TO 194
C     FOLLOW LIST
192 CONTINUE
    MARC = MARC+1
    NAME(MARC) = SLACK
    FROM(MARC) = ISIGN(THD,I)
    COST(MARC) = 0
    CAPAC(MARC) = LEVEL(THD)
    LEVEL(THD) = 0
    FLOOR(MARC) = 0
    NXT = NEXT(THD)
    NEXT(THD) = 0
    THD = NXT
    IF(THD.NE.MNODP1) GO TO 192
194 CONTINUE
    MSLK = MARC
C     ADD EXCESS ARC AT END OF REGULAR SLACKS
    MARC = MARC+1
    FROM(MARC) = ISIGN(MNODP2,-I)
    COST(MARC) = BIG
    CAPAC(MARC) = 0
    FLOOR(MARC) = 0
    NAME(MARC) = XCESS
C
C     LOCATE SOURCES AND SINKS FOR PROBLEM
C ADJUSTED FOR LOWER BOUNDS
C NET = 0
MTREE = 0
THD = MNODP1
DO 200 I200 = 1,MNODE
J = FLOW(I200)
NET = NET+J
IF(J) 30,200,50
C SINK
30 CONTINUE
FLOW(I200) = -J
C LINK DEMANDS IN DECREASING SIZE ORDER
DWN = MNODP1
35 CONTINUE
NXT = DOWN(DWN)
IF(FLOW(NXT)+J.LE.0) GO TO 40
DWN = NXT
GO TO 35
40 CONTINUE
DOWN(DWN) = I200
DOWN(I200) = NXT
LEVEL(I200) = -1
GO TO 200
C SOURCE
50 CONTINUE
MTREE = MTREE+1
ARCID(I200) = -MARC
FLOW(I200) = J
NEXT(THD) = I200
DOWN(I200) = MNODP1
NEXT(I200) = MNODP1
LEVEL(I200) = 1
DUAL(I200) = BIG
THD = I200
200 CONTINUE
C CHECK FOR FEASIBILITY
NSTOP = 16
IF(NET.LT.0) GO TO 999
2331 FORMAT(1X,I5,1H,,4I5,1X,A6)
2332 FORMAT(1X,I5,1H,,5I8,I15)
8339 FORMAT(10H ITERATION,I10)
8704 FORMAT(1H**** DUMP ENABLED AT',I7)
C C ADVANCED START
C C SELECT HIGHEST RANK DEMAND ON LIST
1010 CONTINUE
TO = DOWN(MNODP1)
C IS LIST EXHAUSTED
IF(TO.EQ.MNODP1) GO TO 210
1020 CONTINUE
C SET TO LINK TO ARTIFICIAL
NEWARC = ARTY
NEWPR = BIG
C ANY DEMAND LEFT
IF(FLOW(TO).EQ.0) GO TO 1110
C C LOOK FOR SOURCES FIRST
C
TRY = CAT(TO)
FRM = FROM(TRY)
LST = ISIGN(LNODP1,FRM)
1030 CONTINUE
  
  C     IS IT UNAVAILABLE
  IF(CAPAC(TRY).LE.0) GO TO 1050
  FM = IABS(FRM)
  C     IS IT FROM A NON-SOURCE
  IF(LEVEL(FM).NE.1) GO TO 1050
  IF(ARCID(FM).EQ.ARTY) GO TO 1050
  PRICE = COST(TRY)
  C     IS COST WORSE
  IF(PRICE.GE.NEWPR) GO TO 1050
  C     DOES CAPACITY EXCEED DEMAND
  IF(CAPAC(TRY).GT.FLOW(TO)) GO TO 1040
  C     IS THERE NOT ENOUGH SUPPLY FOR CAPACITY
  IF(FLOW(FM).LT.CAPAC(TRY)) GO TO 1050
  NEWARC = -TRY
  NEWPR = PRICE
  IF(NEWPR.EQ.0) GO TO 1055
  GO TO 1050
1040 CONTINUE
  C     IS THERE NOT ENOUGH SUPPLY FOR DEMAND
  IF(FLOW(FM).LT.FLOW(TO)) GO TO 1050
  NEWARC = TRY
  NEWPR = PRICE
  IF(NEWPR.EQ.0) GO TO 1055
1050 CONTINUE
  TRY = TRY+1
  FRM = FROM(TRY)
  IF(IEOR(FRM,LST).GT.0) GO TO 1030
  IF(NEWARC.EQ.ARTY) GO TO 1070
1055 CONTINUE
  C     ARE WE SENDING DEMAND
  IF(NEWARC.GT.0) GO TO 1060
  C     SEND CAPACITY
  NEWARC = -NEWARC
  FM = IABS(FROM(NEWARC))
  C     GET CAPACITY
  FLW = CAPAC(NEWARC)
  C     MARK UNAVAILABLE
  CAPAC(NEWARC) = -FLW
  C     ADJUST FLOWS
  FLOW(FM) = FLOW(FM) - FLW
  FLOW(TO) = FLOW(TO) - FLW
  GO TO 1020
  C     SEND DEMAND
1060 CONTINUE
  C     MARK UNAVAILABLE
  CAPAC(NEWARC) = -CAPAC(NEWARC)
  C     ADJUST FLOWS
  FM = IABS(FROM(NEWARC))
  FLOW(FM) = FLOW(FM) - FLOW(TO)
  K = BIG
  GO TO 1115
  
  C     LOOK FOR TRANSSHIPMENT POINTS
  
1070 CONTINUE
  TRY = CAT(TO)
FRM = FROM(TRY)
1080 CONTINUE
C IS IT UNAVAILABLE
IF(CAPAC(TRY).LE.0) GO TO 1090
FM = IABS(FRM)
C IS IT ALREADY LINKED
IF(LEVEL(FM).NE.0) GO TO 1090
PRICE = COST(TRY)
C IS COST WORSE
IF(PRICE.GE.NEWPR) GO TO 1090
NEWARC = TRY
NEWPR = PRICE
IF(NEWPR.EQ.0) GO TO 1095
1090 CONTINUE
TRY = TRY+1
FRM = FROM(TRY)
IF(IEOR(FRM,LST).GT.0) GO TO 1080
IF(NEWARC.EQ.ARTY) GO TO 1110
1095 CONTINUE
FM = IABS(FROM(NEWARC))
C DOES CAPACITY EXCEED DEMAND
IF(CAPAC(NEWARC).GT.FLOW(TO)) GO TO 1100
C GET CAPACITY
FLW = CAPAC(NEWARC)
C MARK UNAVAILABLE
CAPAC(NEWARC) = -FLW
C ADJUST FLOWS
FLOW(FM) = FLW
FLOW(TO) = FLOW(TO)-FLW
C LINK IN TO DEMAND LIST
DOWN(FM) = TO
DOWN(MNODP1) = FM
C START NEW CHAIN
LEVEL(FM) = -1
GO TO 1010
1100 CONTINUE
C MARK UNAVAILABLE
CAPAC(NEWARC) = -CAPAC(NEWARC)
C PASS ALONG FLOW DEMAND
FLOW(FM) = FLOW(TO)
C LINK IN NEW DEMAND NODE
DOWN(FM) = DOWN(TO)
DOWN(TO) = FM
DOWN(MNODP1) = FM
NEXT(FM) = TO
ARCID(TO) = NEWARC
LEVEL(FM) = LEVEL(TO)-1
DUAL(TO) = NEWPR
MTREE = MTREE+1
GO TO 1010
C ADD CHAIN TO TREE
1110 CONTINUE
K = 0
1115 CONTINUE
C REMOVE FROM DEMAND LIST
DOWN(MNODP1) = DOWN(TO)
C LINK IN AS LEFTMOST BRANCH
FM = IABS(FROM(NEWARC))
ARCID(TO) = NEWARC
DUAL(TO) = NEWPR
DOWN(TO) = FM
I = NEXT(FM)
NEXT(FM) = TO
J = LEVEL(FM) - LEVEL(TO) + 1
THD = FM
1120 CONTINUE
C MOVE ALONG CHAIN
THD = NEXT(THD)
C ADJUST LEVEL AND DUAL VARIABLES
L = LEVEL(THD)
LEVEL(THD) = L + J
K = K - DUAL(THD)
DUAL(THD) = K
IF(L .NE. -1) GO TO 1120
NEXT(THD) = I
MTREE = MTREE + 1
GO TO 1010
210 CONTINUE
C SET UP TO EXPAND TREE
TO = 1
TRY = ARTP1
FRM = FROM(TRY)
220 CONTINUE
C DO WE NEED TO EXPAND TREE TO REACH ALL NODES
IF(MTREE .EQ. MNODE) GO TO 285
TOO = TO
NEWPR = BIG
C SEARCH FOR LEAST COST CONNECTABLE ARC IN CURRENT TO-GROUP
230 CONTINUE
LVJ = LEVEL(TO)
LST = ISIGN(LNODP1, FRM)
235 CONTINUE
IF(CAPAC(TRY) .LE. 0) GO TO 260
M = COST(TRY)
IF(NEWPR .LT. M) GO TO 260
FM = IABS(FRM)
IF(LEVEL(FM) .EQ. 0) GO TO 240
IF(LVJ .NE. 0) GO TO 260
C TO END IS CONNECTABLE
I = FM
J = TO
K = M
L = TRY
GO TO 250
240 CONTINUE
IF(LVJ .EQ. 0) GO TO 260
C FROM END IS CONNECTABLE
I = TO
J = FM
K = -M
L = -TRY
250 CONTINUE
NEWPR = M
260 CONTINUE
TRY = TRY + 1
FRM = FROM(TRY)
C IS TO-GROUP EXHAUSTED
IF(IEOR(FRM, LST) .GT. 0) GO TO 235
C PREPARE FOR NEXT TO-GROUP
TO = TO + 1
IF(TO.NE.MNODP1) GO TO 270
TO = 1
TRY = ARYP1
FRM = FROM(TRY)
270 CONTINUE
IF(NEWPR.NE.BIG) GO TO 280
IF(TO.NE.TOO) GO TO 230
C NOT ALL NODES CONNECTABLE - CHECK FOR ISOLATED POINTS
DO 275 I275=1,MNODE
IF(LEVEL(I275).NE.0) GO TO 275
C TEST FOR ARCS RUNNING FROM IT
IF(ARCID(I275).EQ.-1) GO TO 274
C CHECK FOR DUMMY - NO ARCS RUNNING TO IT
J275 = CAT(I275)
IF(IABS(FROM(J275)).NE.I275) GO TO 274
WRITE(6,8021) I275
8021 FORMAT(6H NODE ,I10,12H IS ISOLATED)
C ADD ARTIFICIAL
274 CONTINUE
ARCID(I275) = ARY
FLOW(I275) = 0
NEXT(I275) = NEXT(MNODP1)
NEXT(MNODP1) = I275
DOWN(I275) = MNODP1
LEVEL(I275) = 1
DUAL(I275) = -BIG
275 CONTINUE
GO TO 285
280 CONTINUE
ARCID(J) = L
DOWN(J) = I
NEXT(J) = NEXT(I)
NEXT(I) = J
LEVEL(J) = LEVEL(I)+1
DUAL(J) = DUAL(I)-K
NEWARC = IABS(L)
CAPAC(NEWARC) = -CAPAC(NEWARC)
MTREE = MTREE+1
GO TO 220
285 CONTINUE
C CLEAR UPPER BOUND FLAGS ON BASIC ARCS
DO 290 I290=1,MNODE
J290 = IABS(ARCID(I290))
CAPAC(J290) = -CAPAC(J290)
290 CONTINUE
C CLEAR OUT UPPER BOUND FLAGS ON DUMMY ARCS
DO 295 I295=1,MARC
IF(CAPAC(I295)+BIG.EQ.0)
*CAPAC(I295) = 0
295 CONTINUE
C SET UPPER BOUND FOR ARTIFICIAL AND EXCESS
CAPAC(ARY) = BIG
CAPAC(MARC) = BIG
C
C INITIALIZE PRICING
C
TO = 1
TRY = ARYP1
FRM = FROM(TRY)
ITER = 0

Universitas Sumatera Utara
OPTIM = .FALSE.
DMP = .TRUE.

NEW ITERATION

300 CONTINUE
ITER = ITER+1
305 CONTINUE

PRICING

NOTE THAT WE ARE PRICING OUT BASIC ARCS
TOO = TO
NEWPR = 0
310 CONTINUE
PRICE0 = -DUAL(TO)
LST = ISIGN(LNODP1,FRM)
320 CONTINUE
FM = IABS(FRM)
PRICE = DUAL(FM) + PRICE0 - COST(TRY)
IF(CAPAC(TRY)) 325,330,326
325 CONTINUE
PRICE = -PRICE
326 CONTINUE
IF(PRICE.LE.NEWPR) GO TO 330
330 CONTINUE
TRY = TRY+1
FRM = FROM(TRY)
IF(IEOR(FRM,LST).GT.0) GO TO 320
TO = TO+1
IF(TO.NE.MNODP2) GO TO 350
TO = 1
TRY = ARTYP1
FRM = FROM(TRY)
350 CONTINUE
IF(NEWPR.NE.0) GO TO 360
IF(TO.NE.TOO) GO TO 310

OPTIMAL INDICATION

IF(ITDOPT.NE.0) DMP = .TRUE.
OPTIM = .TRUE.
GO TO 795
360 CONTINUE
NEWFRM = IABS(FROM(NEWARC))

RATIO TEST

THETA = IABS(CAPAC(NEWARC))
JTHETA = 0
SET FOR CYCLE SEARCH
CH(2) = ISIGN(LARCP1,CAPAC(NEWARC))
CH(1) = -CH(2)
DW(1) = NEWFRM
DW(2) = NEWTO
LDIFF = LEVEL(NEWFRM) - LEVEL(NEWTO)
KTHETA = 1
IF(LDIFF) 380,450,390
380 KTHETA = 2
390 CONTINUE
DWN = DW(KTHETA)
CHG = CH(KTHETA)
K440 = IABS(LDIFF)
DO 440 I440=1,K440
   IF(IEOR(CHG,ARCID(DWN)).GT.0) GO TO 410
C INCREASING FLOW
   I = IABS(ARCID(DWN))
   POSS = CAPAC(I)-FLOW(DWN)
   JPOSS = -DWN
   GO TO 420
C DECREASING FLOW
410 POSS = FLOW(DWN)
   JPOSS = DWN
C FIND MIN
420 CONTINUE
   IF(THETA.LE.POSS) GO TO 430
   THETA = POSS
   JTHETA = JPOSS
   IF(THETA.EQ.0) GO TO 530
430 CONTINUE
DWN = DOWN(DWN)
440 CONTINUE
DWN(KTHETA) = DWN
C AT COMMON LEVEL
450 CONTINUE
C SEARCH FOR CYCLE END
460 CONTINUE
   IF(DW(1).EQ.DW(2)) GO TO 520
   DO 510 L510=1,2
      DWN = DW(L510)
      IF(IEOR(CH(L510),ARCID(DWN)).GT.0) GO TO 480
C INCREASING FLOW
      I = IABS(ARCID(DWN))
      POSS = CAPAC(I)-FLOW(DWN)
      JPOSS = -DWN
      GO TO 490
C DECREASING FLOW
480 POSS = FLOW(DWN)
   JPOSS = DWN
C FIND MIN
490 CONTINUE
   IF(THETA.LE.POSS) GO TO 500
   THETA = POSS
   JTHETA = JPOSS
   KTTHETA = L510
   IF(THETA.EQ.0) GO TO 530
500 CONTINUE
DWN(L510) = DOWN(DWN)
510 CONTINUE
GO TO 460
520 DWE = DW(1)
C RATIO TEST COMPLETE
530 CONTINUE
   IF(THETA.EQ.0) GO TO 560
C UPDATE FLOWS ON INTACT PORTION OF CYCLE
   DW(1) = NEWFRM
   DW(2) = NEWTO
IF(JTHETA.NE.0)
*DW(KTHETA) = IABS(JTHETA)
DO 550 I550=1,2
DWN = DW(I550)
CHG = ISIGN(THETA,CH(I550))
540 CONTINUE
IF(DWN.EQ.DWE) GO TO 550
FLOW(DWN) = FLOW(DWN)-CHG*ISIGN(1,ARCID(DWN))
DWN = DOWN(DWN)
GO TO 540
550 CONTINUE

IF(JTHETA.NE.0) GO TO 570
CHANGE OF BOUNDS ONLY
CAPAC(NEWARC) = -CAPAC(NEWARC)
GO TO 300
570 CONTINUE
ITHETA = IABS(JTHETA)
IF(JTHETA.GT.0) GO TO 590
J = IABS(ARCID(ITHETA))
SET OLD ARC TO UPPER BOUND
CAPAC(J) = -CAPAC(J)
590 CONTINUE

FLOW ON NEWARC
FLW = THETA
IF(CAPAC(NEWARC).GT.0) GO TO 600
CAPAC(NEWARC) = -CAPAC(NEWARC)
FLW = CAPAC(NEWARC)-FLW
NEWPR = -NEWPR
600 CONTINUE

IF(KTHETA.EQ.2) GO TO 610
Q1 = NEWFRM
Q2 = NEWTO
AID = -NEWARC
NEWPR = -NEWPR
GO TO 620
610 CONTINUE

Q1 = NEWTO
Q2 = NEWFRM
AID = NEWARC
620 CONTINUE

UPDATE TREE

I = Q1
J = DOWN(I)
LSTAR = LEVEL(Q2)+1
IF(THETA.EQ.0) GO TO 730

FLOWS NEED TO BE UPDATED
CHG = ISIGN(THETA,CH(KTHETA))
680 CONTINUE

UPDATE DUAL VARIABLE ON STEM
DUAL(I) = DUAL(I)+NEWPR

UPDATE FLOW ON STEM
N = FLOW(I)
FLOW(I) = FLW

UPDATE ARC ID ON STEM
DIR = ISIGN(1,ARCID(I))
REF = IABS(ARCID(I))
ARCID(I) = AID
C PREPARE FOR LEVEL UPDATES
LSAVE = LEVEL(I)
LDIFF = LSTAR-LSAVE
LEVEL(I) = LSTAR
THD = I
690 CONTINUE
NXT = NEXT(THD)
IF(LEVEL(NXT).LE.LSAVE) GO TO 700
C UPDATE LEVEL
LEVEL(NXT) = LEVEL(NXT)+LDIFF
C UPDATE DUAL VARIABLE
DUAL(NXT) = DUAL(NXT)+NEWPR
THD = NXT
GO TO 690
700 CONTINUE
K = J
710 CONTINUE
L = NEXT(K)
IF(L.EQ.I) GO TO 720
K = L
GO TO 710
720 CONTINUE
C TEST FOR LEAVING ARC
IF(I.EQ.ITHETA) GO TO 790
C PREPARE FOR NEXT UPDATE ON STEM
FLW = N-CHG*DIR
AID = -ISIGN(REF,DIR)
C MOVE DOWN STEM
NEXT(K) = NXT
NEXT(THD) = J
K = I
I = J
J = DOWN(J)
DOWN(I) = K
LSTAR = LSTAR+1
GO TO 680
730 CONTINUE
C ONLY FLOW ON NEW ARC CHANGES
740 CONTINUE
C UPDATE DUAL VARIABLE ON STEM
DUAL(I) = DUAL(I)+NEWPR
C UPDATE FLOW ON STEM
N = FLOW(I)
FLOW(I) = FLW
C UPDATE ARC ID ON STEM
DIR = ISIGN(1,ARCID(I))
REF = IABS(ARCID(I))
ARCID(I) = AID
C PREPARE FOR LEVEL UPDATES
LSAVE = LEVEL(I)
LDIFF = LSTAR-LSAVE
LEVEL(I) = LSTAR
THD = I
750 CONTINUE
NXT = NEXT(THD)
IF(LEVEL(NXT).LE.LSAVE) GO TO 760
C UPDATE LEVEL
LEVEL(NXT) = LEVEL(NXT)+LDIFF
C UPDATE DUAL VARIABLE
DUAL(NXT) = DUAL(NXT)+NEWPR
THD = NXT
GO TO 750

760 CONTINUE
K = J

770 CONTINUE
L = NEXT(K)
IF(L.EQ.I) GO TO 780
K = L
GO TO 770

780 CONTINUE
C TEST FOR LEAVING ARC
IF(I.EQ.I.THETA) GO TO 790
C PREPARE FOR NEXT UPDATE ON STEM
FLW = N
AID = -ISIGN(REF,DIR)
C MOVE DOWN STEM
NEXT(K) = NXT
NEXT(THD) = J
K = I
I = J
J = DOWN(J)
DOWN(I) = K
LSTAR = LSTAR+1
GO TO 740

790 CONTINUE
NEXT(K) = NXT
NEXT(THD) = NEXT(Q2)
NEXT(Q2) = Q1
DOWN(Q1) = Q2
GO TO 300

C SUMMARY
C
795 CONTINUE
INFEAS = .FALSE.
KOST = KOST0
DO 830 I830=1,MNODE
I = IABS(ARCID(I830))
IF((FLOW(I830).NE.0).AND.(COST(I).EQ.BIG)) INFEAS = .TRUE.
KOST = KOST+COST(I)*FLOW(I830)
830 CONTINUE
DO 840 I840=1,MSLK
IF(CAPAC(I840).GE.0) GO TO 840
J840 = -CAPAC(I840)
KOST = KOST+COST(I840)*J840
840 CONTINUE
IF(OPTIM) WRITE(6,8009) ITER
IF(INFEAS) GO TO 850
IF(OPTIM) WRITE(6,8005) KOST
8005 FORMAT(1H+,21X,4HCOST,I16)
GO TO 860

850 CONTINUE
IF(OPTIM) WRITE(6,8008)
8008 FORMAT(1H+,21X,10HINFEASIBLE)
860 CONTINUE
IF(.NOT.OPTIM) GO TO 305
CCCC IF(INFEAS) GO TO 892
WRITE(6,8009) ITER
8009 FORMAT(10H ITERATION,I10)
WRITE(6,8000)
C    MOVE BASIC FLOWS TO CAPACITY CELLS FOR BASIC ARCS
DO 884 I884=1,MNODE
   J884 = IABS(ARCID(I884))
   CAPAC(J884) = -FLOW(I884)
884 CONTINUE
885 CONTINUE
WRITE(6,8023)
8023 FORMAT(10H0ARC FLOWS/
*52H0ARC         FROM   TO           FLOW           COST   )
   TO = 1
   TRY = ARTYP1
   FRM = FROM(TRY)
8886 CONTINUE
   LST = ISIGN(LNODP1,FRM)
8888 CONTINUE
   FLW = MAX0(0,-CAPAC(TRY))+FLOOR(TRY)
   IF(FLW.EQ.0) GO TO 8889
   FM = IABS(FRM)
   WRITE(6,8024) NAME(TRY),FM,TO,FLW,COST(TRY)
8024 FORMAT(1X,I10,1X,2I5,2I15)
   * IF(NAME(TRY).LT.20000) WRITE(7,8025) NAME(TRY)
8025 FORMAT(I6)
8889 CONTINUE
   TRY = TRY+1
   FRM = FROM(TRY)
   IF(IOR(FRM,LST).GT.0) GO TO 8888
   TO = TO+1
   IF(TO.NE.MNODP1) GO TO 8886
   CLOSE(5)
   CLOSE(6)
   STOP 12345
892 CONTINUE
C
C     END OF PROBLEM
C
C
GO TO 5
C
C
EXIT
C
999 CONTINUE
WRITE(6,1999) NSTOP,KARD
1999 FORMAT(5H STOP,I5/I1X,I10,16H DATA CARDS READ)
STOP 77777
END