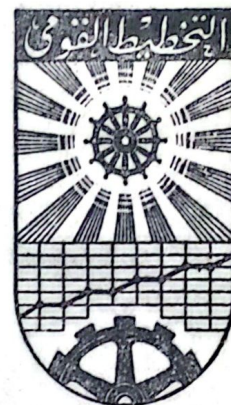


THE ARAB REPUBLIC OF EGYPT

THE INSTITUTE OF NATIONAL PLANNING



Memo. No. (551)

A Course on
Linear Programming and Extensions

Part II

BY

Alwalid N. Elshafei

May 1977

THE CLASSICAL
TRANSPORTATION PROBLEM

Consider the following linear Programming Problem:

$$\begin{array}{ll}
 \text{Minimize } \phi = & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{S.t.} & \sum_{i=1}^m x_{ij} = D_j \\
 & \sum_{j=1}^n x_{ij} = Q_i \\
 & x_{ij} \geq 0
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Minimize } \phi = \\ \text{S.t.} \end{array}} \right\} \begin{array}{ll} \text{DUAL} \\ \text{VARIABLES} \\ \text{PTP} & V_j \\ & U_i \end{array}$$

The following condition has to exist in order to maintain the feasibility of this problem:

$$\sum_{i=1}^m Q_i = \sum_{j=1}^n D_j$$

This condition renders the system of equations in PTP dependent since the sum of the first m equations exactly equals the sum of the remaining n equations. It can be proved that the rank of the system is exactly $(m+n-1)$ and hence

any equation may be called redundant and discarded. The resulting system may be solved as an LP by using the Revised Simplex Method. However, the properties of this system make it possible to handle it in a much easier fashion. It is our purpose here to demonstrate WHY and HOW can this be done.

PROPERTIES OF THE CLASSICAL TRANSPORTATION PROBLEM:

We state, without proof, the following properties.

Proofs are found in any textbook on the subject:

- 1) The subsystem resulting by the exclusion of any redundant equation from the original system will contain an equation in exactly one basic variable.
- 2) FUNDAMENTAL THEOREM OF TRANSPORTATION PROBLEM: The subsystem has the property that every basis is triangular.

In general, a square system is said to be TRIANGULAR if upon suitable rearrangement of its rows and its variables, all coefficients below the diagonal are zero and all coefficients on the diagonal are non-zero. If, in addition, only the diagonal coefficients are non-zero, it is called DIAGONAL.

3) From the definition of basis triangularity, we immediately conclude that:

- a- The subsystem will contain at least one equation having exactly one variable which can be immediately evaluated.
- b- Since there are ONLY $(m+n-1)$ basic variables, then assuming that all non-basic variables are arbitrarily given the value zero, there remains $(m+n-1)$ equation in $(m+n-1)$ unknown. One of these unknowns is determined as in step(a) and it is substituted in the remaining equations, so there will again be at least one equation with one variable, and so forth until all the variables are known and the basic solution is obtained. This is why the FUNDAMENTAL THEOREM IS OF CRUCIAL VALUE. It also follows from (a) and (b) that:
- c- All basic variables take INTEGER VALUES if the row and column totals are integers.

The above properties give the intuitive feeling that there may be a way of obtaining a BASIC SOLUTION of the transportation problem by utilizing the above characteristics which are not necessarily present in a general LP constraints set. We investigate this possibility next.

HOW TO FIND A BASIC FEASIBLE SOLUTION?

Suppose we are given a problem whose data is displayed in table 1 and we trace the following steps:

5	7	10	5	3	5
8	6	9	12	4	10
10	9	8	10	15	10
3	3	10	5	4	

- 1) Consider the cheapest entry in row p (with ties broken on a first in basis), assume it is (p,r). Then put X_{pr}

$$X_{pr} = \min (Q_i, D_j)$$

then we have either of 3 cases:

- i) $Q_i < D_j$, then put $Q_i=0$, $D_j:=(D_j-Q_i)$ and delete i
- ii) $Q_i = D_j$, " " $Q_i=0$, $D_j=0$ and delete i OR j
- iii) $Q_i > D_j$, " " $D_j=0$, $Q_i:=(Q_i - D_j)$ and
delete j.

- 2) Check whether row p has been deleted, if so move to the next row, otherwise repeat step(1), eventually we will get to the situation where one row and one column remain. At this stage both of them are dropped after the last variable has been evaluated.

It is obvious that this process will select as many variables for the basic set as there are rows plus columns, less one, since on the last step when one row and one column remain they are dropped together. Thus we have evaluated $m+n-1$ variables in this process. The following remarks could be proved as theorems:

- 1) Every reduced array retains the property that its row totals are equal to its column totals. So the last remaining variable can acquire a value consistent with the totals for the single row and column still remaining in the final reduced array.
- 2) The variables chosen by the rule for the initial solution constitute a basic set since their values satisfy all the $m+n$ equations and are given UNIQUELY as linear combinations of any $(m+n-1)$ of the $m+n$ marginals $Q_1, \dots, Q_m; D_1, \dots, D_n$.

HOW TO IMPROVE, IF POSSIBLE, THE INITIAL FEASIBLE SOLUTION?

In order to investigate this, let us define the following problem:

$$\begin{array}{ll}
 \text{Maximize } \psi = & \sum_{i=1}^m Q_i U_i + \sum_{j=1}^n D_j V_j \\
 \text{Subject to:} & U_i + V_j \leq c_{ij} \\
 & U_i + V_j \text{ are unrestricted}
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Maximize } \psi = \\ \text{Subject to:} \end{array}} \right\} \begin{array}{l} \text{DUAL} \\ \text{VARIABLES} \\ \text{DTP} \quad X_{ij} \end{array}$$

Clearly, this is the Dual of the standard form of the classical transportation problem. Let us assume that a solution to the DTP is known, thus the vectors U, V and the matrix X are known. From the Primal - Dual we know that the following is true:

- 1) If U, V satisfy the constraints of the dual, then X is an optimal solution and no more improvements are possible.
- 2) If U, V do not satisfy all the constraints, this means that the current solution (U, V) is an infeasible solution to the dual, hence the solution of the primal has to be updated somehow to eliminate such an infeasibility before it becomes a candidate for optimality. Thus,

in fact the P-D(Primal-Dual) relations provide us with an optimality criterion. This, however, requires the calculation of (U,V) corresponding to X . Had the Simplex Method (or one of its variants) been used, then the method of calculating (U,V) would have posed no questions. However, here we do not use the Simplex Method, so how can we calculate U,V ?

CALCULATION OF THE DUAL VARIABLES:

Here we benefit from two criteria in establishing a quick method for calculating the dual variables, they are:

- a) Basis Triangularity
- b) P-D Relations (AGAIN ?!!)

In fact the way these criteria interact in the present context is very elegant. Let us see how we proceed

Firstly recall that only $(m+n-1)$ of the X_{ij} 's are in the basis and are generally (i.e. unless degeneracy occurs) non-zero. From the P-D relations we know that for each of these X 's the corresponding inequality is automatically satisfied as an equality.

$\therefore m+n-1$ of the inequalities $U_i + V_j \leq C_{ij}$ are in fact equalities and their i 's and j 's are known. These equalities form an independent set of equations since they correspond to the elements of a basic feasible solution .

Secondly The number of u 's and v 's involved in this system is $(m+n)$, but since there are only $(m+n-1)$ independent equations, one of these variables can take any arbitrary value (say ZERO for simplicity).

So, We know which equalities form the set, we choose one of the variables and assign the value zero to it, and we are left with $(m+n-1)$ equations in $(m+n-1)$ unknowns. We can thus calculate the elements of u and v .

ONCE u and v ARE KNOWN:

We substitute the appropriate values in the residual set of inequalities [which had remained when the $(m+n-1)$ equalities were taken out of the original set] and if all the inequalities are satisfied, this means that u, v is a FEASIBLE SOLUTION TO THE DUAL and thus X is OPTIMAL, otherwise, this INFEASIBILITY HAS TO BE REMOVED and so X can not

be considered as a candidate for optimality, in terms of the dual problem, yet.

An interesting question emerges:

Do we really have to solve $(m+n-1)$ equations in $(m+n-1)$ unknowns in the usual manner (inversion, elimination, ... etc.) to get the values of the dual variables?

The answer to this question is (FORRUNATELY) NO . THE P-D RELATIONS relieve such burden as we now show.

NOTICE that the resulting system in the PTP is the dual of the system $\{u_i + v_j = C_{ij} \quad \forall i, j \mid X_{ij} \text{ is basic}\}$. We know that the former system has certain properties which allow its variables to be calculated by repetitive scanning. One would expect that the variables in the latter system can be also calculated by repetitive scanning. This is in fact true and all the $(m+n-1)$ dual variables are calculated by repetitive scanning.

HOW TO IMPROVE AN EXISTING SOLUTION:

Since we are concerned with the violated inequalities in the dual problem, it seems appropriate to choose one of them (Unfortunately we cannot choose more than one at a

time, CAN YOU SHOW WHY?) and change X so that this inequality is not any longer violated. A non-ambiguous way of doing so is by forcing it to be a strict equality. Again from the P-D relations this is achieved by **choosing** the X_{ij} corresponding to this inequality and **entering it in the basis**.

HOW TO ENTER X_{ij} IN THE BASIS?

Once it is decided which non-basic variable should be entered into the basis, it is our main concern that this should be done WITHOUT UPSETTING THE FEASIBILITY OF THE PRIMAL.

We recall that in the Simplex Method, this is done by choosing the pivot row which has the appropriate quotient. However here we do not use the Simplex Method, so how can we achieve the same purpose?

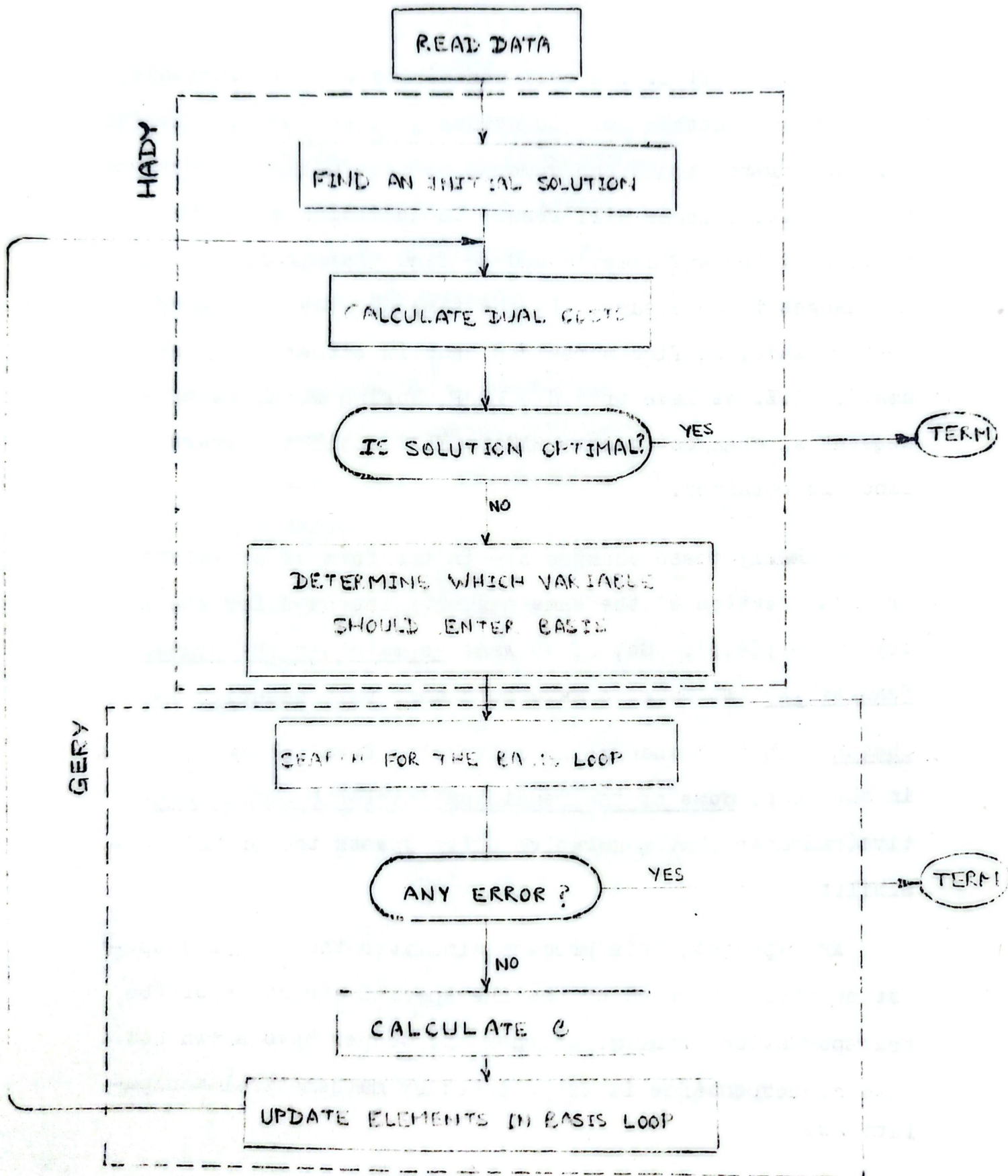
THE NOTION OF BASIS LOOP AND ITS USE IN UPDATING THE BASIC SOLUTION:

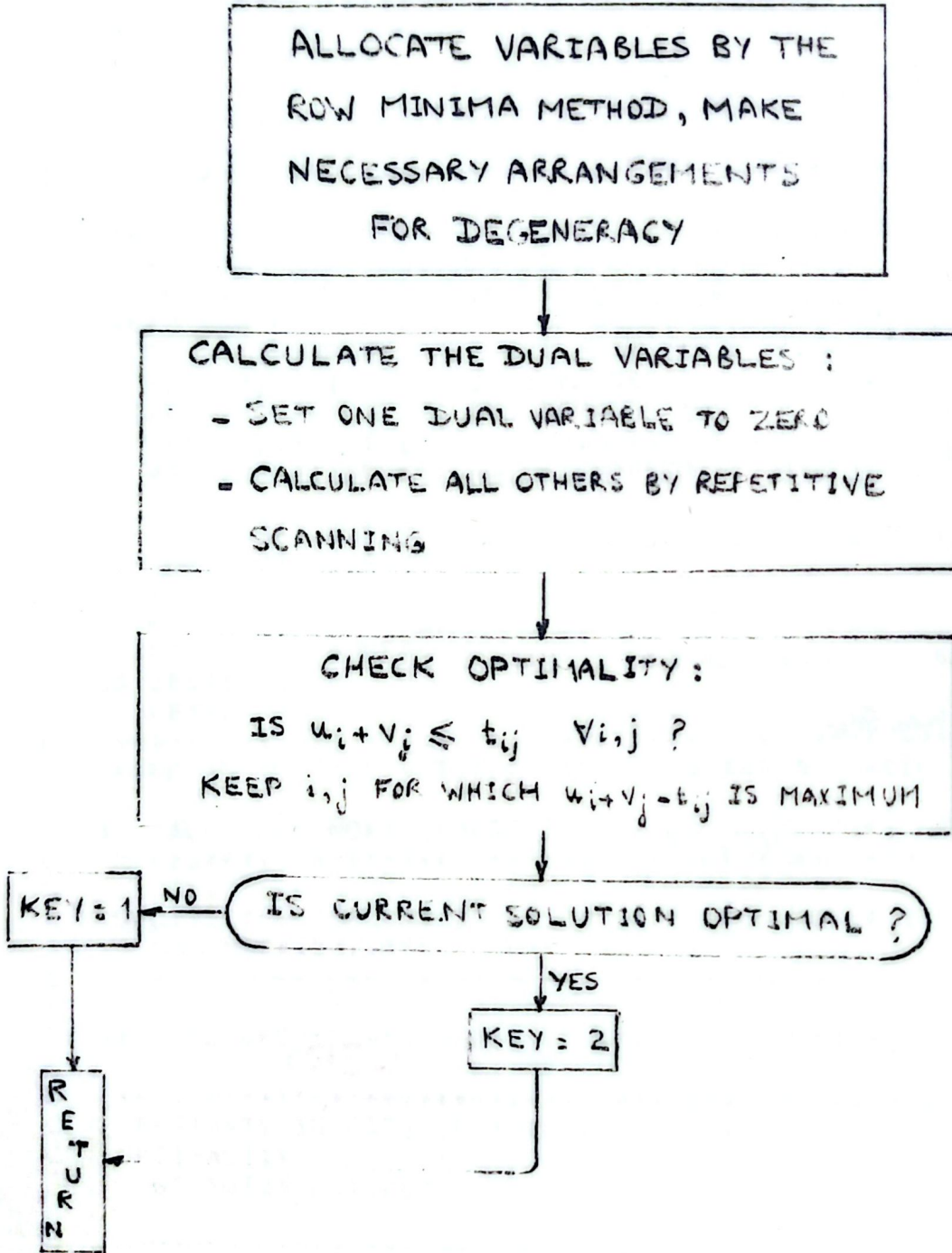
In order to update the basic solution, we benefit from the special structure of the transportation problem. For example, we know that if a certain quantity is introduced in cell(i,j), then the same quantity has to be subtracted

from a basic cell(i.e. a cell containing a basic variable) in row i and another cell in column j to balance the availability of **source** i and the demands of destination j. However, these latter changes will result in upsetting some other availabilities and demands and we find that in fact a CHAIN of changes is necessary. By the time all the necessary changes are made, we find ourselves back in either row i or column j. I.E. we have traced a LOOP, during which we made sequential changes in some BASIC CELLS until the overall balance is retained.

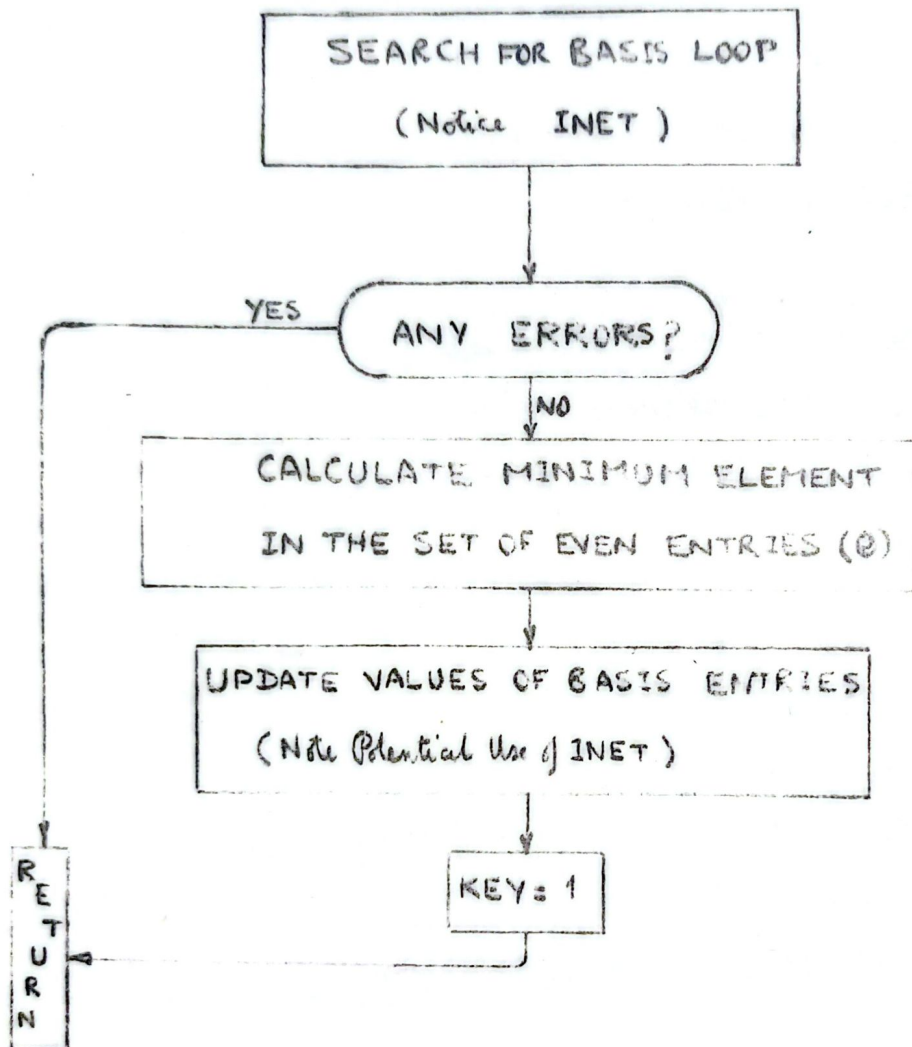
Obviously these changes are in the form of an addition or a subtraction of the same quantity intended for the entry in cell(i,j). So, if we want to maintain the primal feasibility, we have to make sure that this quantity is chosen such that when it is subtracted from the basic cells in the loop, none of the remaining entries becomes negative(remember that a negative entry upsets the primal feasibility).

As expected, this process simulates the pivoting operation when performed on the the special structure of the transportation problem. Charnes and Cooper have shown how this correspondance is established by mathematical manipulations.





HADY



GERY

```

C *****
C A PROGRAM FOR THE SOLUTION OF THE TRANSPORTATION PROBLEM BY
C THE MOD1 METHOD
C *****
C   DIMENSION IORIG(10),IDEST(10),ICOST(10,10),JBAS(10,10)
C   DIMENSION KORIG(10),KDEST(10)
C *****
C   *READ PROBLEM NUMBER , NUMBER OF ORIGINS , NUMBER OF DESTINATIONS
C   **AND THE CONTINUATION PARAMETER
C   5 READ(2,1)NUM,NORIG,NDEST,MORE
C   READ THE COST MATRIX
C   DO 10 I=1,NORIG
10  READ(2,1)(ICOST(I,J),J=1,NDEST)
C   READ AVAILABILITIES AND DEMANDS
C   READ(2,1)(IORIG(I),I=1,NORIG)
C   READ(2,1)(IDEST(J),J=1,NDEST)
C
C   DO 20 J=1,NDEST
20  KDEST(J)=IDEST(J)
C   DO 30 I=1,NORIG
C   KORIG(I)=IORIG(I)
C   DO 30 J=1,NDEST
30  JBAS(I,J)=0
C   KEY=2
C *****
C   FIND AN INITIAL FEASIBLE SOLUTION AND/OR CHECK OPTIMALITY
C
C   35 CALL HADY(NORIG,NDEST,ICOST,IORIG,IDEST,K1,NROW1,NCOL1,JBAS,KEY)
C *****
C   UPDATE THE CURRENT SOLUTION OR TERMINATE IF IT IS OPTIMAL
C   GO TO(36,38),KEY
C   -----
C   36 CALL GERY(NORIG,NDEST,ICOST,IORIG,IDEST,K1,NROW1,NCOL1,JBAS,KEY)
C
C *****
C   TERMINATE IN CASE OF INTERNAL ERROR IN GERY , OTHERWISE CHECK
C   OPTIMALITY
C   GO TO(35,60),KEY
C
C   -----
C   SUBTRACT K1 FROM ALL BASIC VARIABLES (CHECK THE REASON FOR DOING SO
C   IN HADY)
C   38 DO 39 I=1,NORIG
C   DO 39 J=1,NDEST
C   IF(JBAS(I,J))39,39,37

```

```
37 JBAS(I,J)=JBAS(I,J)-K1
39 CONTINUE
```

C
C

C PRINT FINAL RESULTS IN CASE THERE IS A FINAL SOLUTION

```
WRITE(3,6)NUM
```

```
DO 50 I=1,NORIG
```

```
50 WRITE(3,1)(JBAS(I,J),J=1,NDEST),KORIG(I)
```

```
WRITE(3,1)(KDEST( J),J=1,NDEST)
```

C

```
60 IF(MORE)100,100,5
```

```
100 CALL EXIT
```

C

```
1 FORMAT(10X,12I5)
```

```
6 FORMAT(1H1,'RESULTS FOR PROBLEM NO.',I5/)
```

C

```
*****  
END
```



```
C *****
C A SUBROUTINE TO FIND AN INITIAL SOLUTION OF A TRANSPORTATION PROBLEM
C USING THE METHOD OF ROW MINIMA
C APPLICABLE TO DEGENERATE CASES BUT NOT TO UNBALANCED ONES
C ALL X'S HAVE TO BE UPDATED BY SUBTRACTING K1
C *****
C
C SUBROUTINE HADY(NORIG,NDEST,ICOST,IORIG,IDEST,K1,NROW1,NCOL1,JBAS,
C *KEY)
C -----
C DIMENSION IU(10),IV(10),JBAS(10,10),IORIG(10),IDEST(10)
C DIMENSION ICOST(10,10),IU1(10),IV1(10)
C *****
C GO TO(320,5),KEY
C -----
C INITIALIZE
C 5 K1=3000
C NDST1=NDEST
C NORG1=NORIG
C NCOL=1
C NROW=1
C ISAV=K1
C DO 140 J=1,NDEST
140 IV(J)=1
C FROM HERE UNTIL JUST BEFORE 320 THE VARIABLES ARE ALLOCATED
150 IF(ICOST(NROW,NCOL)-ISAV)180,160,160
160 NCOL=NCOL+1
C IF(NCOL-NDEST)150,150,200
180 IF(IV(NCOL)-1)160,190,160
190 ISAV=ICOST(NROW,NCOL)
C NCOL1=NCOL
C GO TO 160
C
C 200 IF(IORIG(NROW)-IDEST(NCOL1))210,230,310
210 ISET=IORIG(NROW)+K1
C
C JBAS(NROW,NCOL1)=ISET
C
C IDEST(NCOL1)=IDEST(NCOL1)-IORIG(NROW)
C GO TO 290
C
C 230 ISET=IORIG(NROW)+K1
C
C JBAS(NROW,NCOL1)=ISET
C
C IV(NCOL1)=0
C MAKE NECESSARY ARRANGEMENTS FOR DEGENERACY
C NCOL=1
C ISAV=K1
C IF(NROW-NORIG)240,320,320
```

```

240 IF(ICOST(NROW,NCOL)-ISAV)250,270,270
250 IF(IV(NCOL)-1)270,260,270
260 ISAV=ICOST(NROW,NCOL)
    NCOL1=NCOL
270 NCOL=NCOL+1
    IF(NCOL-NDEST)240,240,280
C
280 JBAS(NROW,NCOL1)=K1
C
290 NROW=NROW+1
    IF(NROW-NORIG)300,300,320
300 NCOL=1
    ISAV=K1
    GO TO 150
C
310 ISET=IDEST(NCOL1)+K1
C
    JBAS(NROW,NCOL1)=ISET
C
    IDORIG(NROW)=IDORIG(NROW)-IDEST(NCOL1)
    IV(NCOL1)=0
    GO TO 300
C *****
C START CALCULATING THE DUAL VARIABLES
C -----
C A) INITIALIZE AS FOLLOWS
C * SET ALL DUAL VARIABLES IU & IV EQUAL TO ZERO
C * FLAG ALL DUAL VARIABLES, IU1 & IV1 ARE SET EQUAL TO ZERO
320 DO 330 I=1,NORIG
    IU(I)=0
330 IU1(I)=0
    DO 340 J=1,NDEST
    IV(J)=0
340 IV1(J)=0
C UNFLAG THE FIRST ROW
    IU1(1)=1
    NROW=1
    NCOL=1
C B) FROM HERE UNTIL JUST BEFORE 540 THE REMAINING DUAL VARIABLES ARE
C DETERMINED BY REPETITIVE SCANNING
360 IF(JBAS(NROW,NCOL)-K1)380,370,370
370 IV(NCOL)=ICOST(NROW,NCOL)-IU(NROW)
C * UNFLAG COLUMN NCOL
    IV1(NCOL)=1
380 NCOL=NCOL+1
    IF(NCOL-NDEST)360,360,400
400 NROW=NROW+1
    IF(NROW-NORIG)420,420,500
420 IF(IU1(NROW)-1)430,400,430
430 NCOL=1

```



```

440 IF(JBAS(NROW,NCOL)-K1)450,470,470
450 NCOL=NCOL+1
    IF(NCOL-NDEST)440,440,400
470 IF(IV1(NCOL))480,450,480
480 IU(NROW)=ICOST(NROW,NCOL)-IV(NCOL)
C  * UNFLAG ROW NROW
    IUI(NROW)=1
    NCOL=1
    GO TO 360
C  * CHECK WHETHER ANY ROWS ARE STILL FLAGGED , IF SO RETURN TO 430 ,
C  OTHERWISE GO TO 540
500 NROW=1
510 IF(IUI(NROW))520,430,520
520 NROW=NROW+1
    IF(NROW-NORIG)510,510,540
C  ++++++
C  CHECK OPTIMALITY OF THE CURRENT SOLUTION
C  -----
540 ISAV=0
    NROW=1
550 NCOL=1
560 IF(JBAS(NROW,NCOL)-K1)620,570,570
570 NCOL=NCOL+1
    IF(NCOL-NDEST)560,560,590
590 NROW=NROW+1
    IF(NROW-NORIG)550,550,610
610 IF(ISAV)700,800,700
620 IS1=IU(NROW)+IV(NCOL)-ICOST(NROW,NCOL)
    IF(IS1-ISAV)570,570,640
640 ISAV=IS1
C
    NROW1=NROW
    NCOL1=NCOL
    GO TO 570
C  ++++++
C  CURRENT SOLUTION IS NON-OPTIMAL
C  -----
700 KEY=1
    RETURN
C  ++++++
C  CURRENT SOLUTION IS OPTIMAL
C  -----
800 KEY=2
    RETURN
C
C  ++++++
C  END

```



```

C *****
C A SUBROUTINE TO IDENTIFY BASIS LOOP AND UPDATE ITS ENTRIES
C *****
C SUBROUTINE GERY(NORIG,NDEST,ICOST,IORIG,IDEST,K1,NROW1,NCOL1,JBAS,
C *KEY)
C -----
C DIMENSION ICOST(10,10),JBAS(10,10),IORIG(10),IDEST(10)
C DIMENSION INET(040),INET1(10),INET2(10)
C *****
C A) INITIALZE
C * SET K2 .GE. (10*K1)
C K2=30000
C K=NORIG+NDEST
C K=2*K
C DO 20 I=1,K
20 INET(I)=0
C ** THUS THE SIZE OF ARRAY INET .EQ. 2*(NORIG+NDEST) **
C DO 40 I=1,NORIG
40 INET1(I)=0
C DO 60 I=1,NDEST
60 INET2(I)=0
C *****
C B) START SEARCHING FOR THE BASIS LOOP
C -----
C ** FROM HERE UNTIL JUST BEFORE 900 OUR MAIN TASKS ARE
C 1) RECOGNIZING EACH ELEMENT OF LOOP AND PUTTING ITS ROW AND
C COLUMN NUMBERS IN TWO ADJACENT LOCATIONS OF ARRAY INET
C 2) DETECT ANY UNEXPECTED ERRORS
C -----
C I=1
C INET(I)=NROW1
C I1=I+1
C INET(I1)=NCOL1
C NROW=NROW1
C NCOL=1
C I=I+2
100 IF(JBAS(NROW,NCOL)-K1)160,120,120
120 IF(NCOL-NCOL1)140,160,140
140 INET(I)=NROW
C I1=I+1
C INET(I1)=NCOL
C I=I+2
C GO TO 220

```

```

160 NCOL=NCOL+1
   IF(NCOL-NDEST)100,100,200
200 I=1
C   *) NO BASIS ELEMENT IN THE ROW OF THE ENTERING , ERROR TYPE 1
   WRITE(3,1)I,NROW1,NCOL1
   KEY=2
C
   RETURN
C   ++++++
220 INET2(NCOL)=1
   NROW=1
240 IF(JBAS(NROW,NCOL)-K1)260,380,380
260 NROW=NROW+1
   IF(NROW-NORIG)240,240,300
300 I=I-2
   IF(I)360,360,340
340 NROW=INET(I)
   I1=I+1
   NCOL=INET(I1)
   INET2(NCOL)=0
C
   GO TO 520
C   *) THERE IS NO BASIS LOOP
360 I=2
C
   WRITE(3,2)I,NROW1,NCOL1
   KEY=2
C
   RETURN
C   ++++++
380 I2=I-2
   IF(NROW-INET(I2))400,260,400
400 IF(INET1(NROW))300,420,300
420 INET(I)=NROW
   I1=I+1
   INET(I1)=NCOL
   I=I+2
   IF(NROW-NROW1)480,460,480
460 I=I-2
   GO TO 300
480 INET1(NROW)=1
   NCOL=1
500 IF(JBAS(NROW,NCOL)-K1)520,560,560
520 NCOL=NCOL+1
   IF(NCOL-NDEST)500,500,640
560 I1=I-1
   IF(NCOL-INET(I1))580,520,580
580 IF(INET2(NCOL))640,600,640
600 INET(I)=NROW
   I1=I+1

```



```

      INET(I1)=NCOL
      I=I+2
      IF(NCOL-NCOL1)220,700,220
640  I=I-2
      IF(I)360,360,680
680  NROW=INET(I)
      I1=I+1
      NCOL=INET(I1)
      INET1(NROW)=0
      GO TO 260
C
700  I=3
C  *) CALCULATE THE MINIMUM ENTRY IN BASIS LOOP
      ISAV=K2
720  NROW=INET(I)
      I1=I+1
      NCOL=INET(I1)
      IF(JBAS(NROW,NCOL)-ISAV)760,780,780
760  ISAV=JBAS(NROW,NCOL)
      NROW2=NROW
      NCOL2=NCOL
780  IF(NCOL-NCOL1)800,860,800
800  I=I+4
      IF(I-K)720,720,840
C  *) THE BASIS LOOP HAS MORE ENTRIES THAN THER ARE IN THE BASIS ,
C  ERROR TYPE 3
840  I=3
C
      WRITE(3,3)I,NROW1,NCOL1
      KEY=2
C
      RETURN
C  ++++++
860  IF(ISAV-K2)900,880,880
C  *) THERE IS NO ELEMENT IN THE BASIS LOOP .LT. K2, ERROR TYPE 4
880  I=4
C
      WRITE(3,4)I,NROW1,NCOL1
      KEY=2
C
      RETURN
C  ++++++
C  C) NOW THE LOCATIONS OF ALL ELEMENTS IN THE BASIS ARE SEQUENTIALLY
C  ARRANGED (PAIRWISE) IN ARRAY INET AND ISAV CONTAINS THE MINIMUM
C  ENTRY IN THE LOOP , SO WE CAN UPDATE THE BASIC SOLUTION BY
C  SUCCESSIVE ADDITIONS AND SUBTRACTIONS FROM THE ELEMENTS OF THE
C  BASIS LOOP
C  -----
900  J=-1
      NROW=INET(1)

```



```

NCCL=INET(2)
JBAS(NROW,NCOL)=ISAV
ISAV=ISAV-K1
C
  I=3
920 NROW=INET(I)
  I1=I+1
  NCOL=INET(I1)
  IF(NROW-NROW2)1000,960,1000
960 IF(NCOL-NCOL2)1000,980,1000
980 JBAS(NROW,NCOL)=0
  GO TO 1020
C
1000 ISET=JBAS(NROW,NCOL)+J*ISAV
  JBAS(NROW,NCOL)=ISET
C
1020 J=-J
  I=I+2
  IF(NCOL-NCOL1)920,1080,920
C *****
C NOW THAT THE NEW BASIC SOLUTION HAS BEEN OBTAINED , SET KEY = 1 TO
C ENABLE HADY TO CHECK OPTIMALITY
C -----
1080 KEY=1
  RETURN
C
1 FORMAT(10X,'ERROR TYPE 1'/15X,'NO BASIS ELEMENT IN ROW WHICH CONTA
*INS ENTERING ELEMENT IN AN ITERATION ',3I3)
2 FORMAT(10X,'ERROR TYPE 2'/15X,'THERE IS NO BASIA LOOP',3I3)
3 FORMAT(10X,'ERROR TYPE 3'/15X,'THE BASIS LOOP HAS MORE ELEMENTS TH
*AN THERE ARE IN THE BASIS',3I3)
4 FORMAT(10X,'ERROR TYPE 4'/ 15X,'THERE IS NO ELEMENT IN THE BASIS L
*OOP LESS THSN K2',3I3)
C *****
C END

```

A MANUAL TRACE OF DETERMINING BASIS LOOP AND 0
BY 'GERY', GIVEN A CURRENT BF SOLUTION

⑥ 3004	⑤ 3003		
	④ 3003	③ 3008	
① *		② 3001	3007

7 NET1= 0 1

11 NET1= 0 1

8 NET1= 0

4 6 9 7
NET2=0 NET2=0 NET2=0 NET2=0
1 1

(1)	(2)
I=1	240 JBAS(1,3) < K1 → 260
INET(1)=3	260 NROW=2 < NORIG → 240
I1=2	240 JBAS(2,3) > K1 → 380
INET(2)=1	380 I2=I-2=3
NROW=3	{NROW-INET(3)} =
NCOL=1	=2-3 i.e. -ve → 400
I=3	400 INET1(2)=0 → 420
100 JBAS(3,1) > K1 → 120	420 INET(5)=2
120 NCOL=NCOL1=1 → 160	INET(6)=3
160 NCOL=2	I=7
NCOL < NDEST → 100	NROW < NORIG → 480
100 JBAS(3,2) < K1 → 160	480 INET1(2)=1
160 NCOL=3	NCOL=1
NCOL < NDEST → 100	500 JBAS(2,1) < K1 → 520
100 JBAS(3,3) > K1 → 120	520 NCOL=2
120 NCOL ≠ NCOL1 → 140	NCOL < NDEST → 500
INET(3)=3	500 JBAS(2,2) > K1 → 560
INET(4)=3	560 I1=6


```

I=5... → 220
INET(3)=1
NROW=1

(3)

600 INET(7)=2
    INET(8)=2
    I=9
    NCOL ≠ NCOL1 → 220
220 INET2(2)=1
    NROW=1
240 JBAS(1,2) > K1 → 380
380 I2=7
    NROW-INET(7)=
    =1-2, i.e.-ve → 400
400 INET(1)=0 → 420
420 INET(9)=1
    INET(10)=2
    I=11
    NROW ≠ NROW1 → 480
480 INET1(1)=1
    NCOL=1
500 JBAS(1,1) > K1 → 560
560 I1=10
    NCOL-INET(10)=
    =1-2, i.e.-ve → 580
580 INET2(1)=0 → 600
600 INET(11)=1
    INET(12)=1
    I=13
    NCOL=NCOL1 → 700
END OF BASIS LOOP TRACE
700 I=3

```

```

NCOL-INET(6)=
    =2.3, i.e.-ve → 580
580 INET2(2)=0 → 600

(4)

ISAV=∞
720 NROW=INET(3)=3
    I1=4
    NCOL=INET(4)=3
    JBAS(3,3) < ISAV → 760
760 ISAV=JBAS(3,3)=1
    NROW2=3
    NCOL2=3
780 NCOL ≠ NCOL1 → 800
800 I=I+4=7
    I < K → 720
720 NROW=INET(7)=2
    NCOL=INET(8)=2
    JBAS(3,3) > ISAV → 780
780 NCOL ≠ NCOL1 → 800
800 I=I+4=11
    I < K → 720
720 NROW=INET(11)=1
    NCOL = INET(12)=1
    JBAS(1,1) > ISAV → 780
780 NCOL=NCOL1 → 860
END OF CALCULATION

```

TREATMENT OF SOME SPECIAL
CASES AS TRANSPORTATION PROBLEMS

In order to be able to solve a LP, with TP structure, by the MODI method or any other similar method, the following conditions are required to prevail:

- 1) Total Supply equals Total Demand
- 2) All entries in the cost matrix are feasible
- 3) No capacity restrictions are imposed on the flows or any routes.

However, some situations may have a first hand formulation which does not comply with one(or more) of the above restrictions. In some cases it **may be** possible to modify the formulation so as to comply with the conditions as we investigate in the sequel.

A) SUPPLY EXCEEDS DEMAND:

This case is easily remedied by adding a dummy destination whose demand is given by $D_{m+1} = \sum_{i=1}^m Q_i - \sum_{j=1}^n D_j$. This dummy destination swallows the excess capacity in the system. In order to complete the formulation, we need to calculate the cost entries in the resulting extra column

(n+1). These costs are, physically, the costs of underutilizing the capacities at the corresponding sources and it is up

	1	2	3	4	DUMMY	
A	12	6	7	8	0	700
B	11	3	10	5	0	600
C	7	9	4	11	0	1200
	300	200	900	400	700	

to the user to set it to whichever appropriate figure; figures can vary from source to source. It is obvious that these figures govern the final solution to the problem. Generally, it has been convenient to set all these entries to zero, unless a better way of calculating them exists.

B) DEMAND EXCEEDS SUPPLY:

Generally, this case can also be remedied by adding a dummy source whose capacity is given by $Q_{m+1} = \sum_{j=1}^n D_j - \sum_{i=1}^m Q_i$. The dummy source acts as a fictitious source for supplying the demand which is in excess of the system's capacity.

The cost entries in the additional row(m+1) are the penalties for

	1	2	3	4	
A	12	6	7	8	500
B	11	3	10	5	300
C	7	9	4	11	700
DUMMY	?	?	?	?	300
	300	200	900	400	

not supplying the customers and it is up to the user to choose the appropriate figures for these entries.

C) DEMAND EXCEEDS SUPPLY WITH ADDITIONAL RESTRICTIONS ON SHORTAGE:

Suppose that we meet the above situation and we are told that the costs of undersupplying the customers are 4,6,9 & 2. We are required to work out a plan in which no customer is undersupplied by more than 25% of its demand and the overall cost is minimized.

The resulting formulation is displayed on the following page. In this case we need n dummy sources, the capacity of each is given by $Q_{m+j} = (0.25 \times D_j)$.

If, in the resulting formulation (in cases like this), we have:

	1	2	3	4	DUMMY	
A	125 ₁₂	200 ₆	175 ₇		8	500
B	11	3	10	300 ₅	∞	300
C	100 ₇	9	600 ₄	11		700
1	75 ₄		∞		0	75
DUMMIES2					50 ₀	50
3		6	125 ₉		100 ₀	225
4	∞			100 ₂	0	100
	300	200	900	400	150	

$$\left(\sum_{i=1}^m Q_i + 0.25 \sum_{j=1}^n D_j \right) < \sum_{j=1}^n D_j, \text{ it means}$$

that the problem does not have a feasible solution, i.e. it is impossible to find a solution in which the shortage at any destination does not exceed 25% of its demand.

Thus, in general, we check whether $\sum_{i=1}^m Q_i \geq (1-p) \sum_{j=1}^n D_j$ (where p is the upper bound on the shortage allowed at any destination ^{*}) or not. If not, the above condition is $\sum_{i=1}^m Q_i \geq \sum_{j=1}^n (1-p_j) D_j$.

formulation does not have a feasible solution. Generally, we need a dummy destination with capacity D_{n+1} given by:

$$D_{n+1} = \sum_{i=1}^m Q_i - \left\{ (1-p) \sum_{j=1}^n D_j \right\}$$

A suitable cost arrangement is shown in the table together with a feasible(BUT NOT NECESSARILY OPTIMUM) solution.

D) TREATMENT OF IMPOSSIBLE ROUTES:

Any infeasible source-destination combination can be represented by an entry in the TP tableau with infinite cost(K). Thus the resulting problem can be solved as a classical transportation problem. It is obvious that an entry with infinite cost will not be present in the optimum solution unless there is not a single solution which is better, which physically means that no feasible solution exists to the original problem. Such a solution is easily recognized as it has an infinite cost(resulting from the entry with infinite cost). However, any such infeasibility is detected at termination rather than at the end of what may be equivalent to PHASE I in LP.

From the computational point of view, it was found convenient to take $K=10 \times \left\{ \max_{i,j} t_{ij} \right\} \times \max_{i,j} (Q_i, D_j)$ and not necessarily ∞ .

Consider the following example :

	1	2	3	4					
A	200	12	100	6	300	7	200	8	700
B	150	11	200	3	500	10	200	5	600
C	200	7	300	9	200	4	400	11	500
	300	200	900	400					

UB_{ij}	t_{ij}
-----------	----------

a typical cell

	1	2	3	4	S_1	S_2	S_3	S_4	UB_{ij}
x_{11}		12			0				200
x_{12}			6			0			100
x_{13}				7			0		300
x_{14}					8			0	200
x_{21}		11			0				150
x_{22}			3			0			200
x_{23}				10			0		500
x_{24}					5			0	200
x_{31}		7			0				200
x_{32}			9			0			300
x_{33}				4			0		200
x_{34}					11			0	400
	300	200	900	400	250	400	100	400	

The problem can be solved as an enlarged transportation problem, the construction of which is exhibited on the **previous** page. We need n slack columns, each one representing the sum of the slack capacities leading to a destination⁺. The capacity of column S_j is given by : $S_j = \sum_{i=1}^m UB_{ij} - D_j$. Obviously, if any of the S_j 's is found to be negative, this means that the original problem is infeasible. Notice the physical interpretation of a negative S_j .

Notice that from a computational point of view, whenever any route (i,j) does not have any upper bound on the shipments through it, it suffices to put the corresponding $UB_{ij} = \max (Q_i, D_j)$ rather than ∞ .

EXERCISE: Solve the following problem once as a TP and once as an LP and compare the CPU times (solution times only, i.e. no compilation or link-editing times are to be included). Arrange the output of the LP to appear in a matrix form. Notice that the cost matrix of the enlarged problem can be generated from that of the original problem by a trivial MG. The same applies to availabilities and demands.

-
- + An alternative procedure where there are m slack columns is possible. In this case, each column represents slack capacities on the routes from source i .

Remember ~~that it is useful~~ to incorporate feasibility tests in such problems whenever possible. It is true that if the Revised Simplex is used to solve the LP, it will discover infeasibility (if it existed) at the end of phase I. However, since we have a problem oriented test which is easily to perform, why not incorporate it in the program we write to solve this problem as an LP. This will save not only the computation time for Phase I, but also that for the MG. To take this one step further, if this problem is put to an OR man and he performs such quick and easy test, it might save him the effort of writing and debugging the MG in the first place!! . In general, it pays to look for such tests to incorporate in a problem solving procedure as they usually result in substantial effort savings and consequently an efficient solution procedure.

1 2 3 4 5 6 7 8 9 10

A	20	10	20	100	50	90	10	45	15	65	11	60	0	∞	12	20	16	60	50	90	147
B	37	55	25	80	12	20	7	19	8	23	16	15	15	33	9	40	0	∞	13	10	40
C	48	∞	40	44	30	95	5	60	27	30	5	66	18	25	17	10	10	50	9	15	172
D	55	20	5	77	28	78	0	∞	22	65	8	50	20	40	13	20	23	30	8	25	85
E	30	82	10	22	10	80	9	75	34	55	21	70	16	∞	4	30	20	55	20	40	106

A Capacitated Transportation Problem

REFERENCES

- (1) Wagner, H.M., "On a class of Capacitated Transportation Problems," Management Science, "Vol. 5, 1959, pp. 304-318

THE TRANSPORTATION ON PROBLEM
WITH PRODUCTION AND OVERHEAD COSTS

Three factories A,B,C produce a certain commodity which is required at five destinations. Together with transportation costs(C_{ij} 's), there are also processing costs(per unit) λ_i 's at the factories. It is also known that there are fixed overhead costs(i.e. independent of the volume of output) F_i 's at the factories. Generally, it is required to decide upon the transportation programme so as to minimize the total cost of manufacturing and transportation. Here we consider two cases:

A) SUPPLY EQUALS DEMAND:

Consider the following example:

FACTORY	OVERHEAD	PROCESSING COST	DESTINATIONS					AVAILAB'S
			1	2	3	4	5	
A	6000	3	5	7	10	5	3	500
B	5000	4	8	6	9	12	4	1000
C	7000	2	10	9	8	10	15	1000
DEMANDS			300	300	1000	500	400	

Each unit produced at factory i and shipped to destination j will incur a combined cost of $(\lambda_i + C_{ij})$. To satisfy demands, all the three factories have to be in operation, so all overheads are going to be charged. However, the shipping schedule has to be worked out. This is accomplished by replacing each entry (C_{ij}) in the tableau by $(C_{ij} + \lambda_i)$ and solving the resulting problem as a classical transportation problem.

B) SUPPLY EXCEEDS DEMAND:

Suppose that in the previous example, capacities of A, B and C were 1500, 1200 and 1300. Notice that the total demand equals 2500, so in fact not all the factories need to be in operation as any of the following combinations can fulfill all the demands: (A,B), (B,C), (A,C) and (A,B,C). In this case a choice has to be made so as to choose the combination **yielding** the least possible total cost (Overheads + Processing + Transportation).

This problem is known as the FIXED CHARGE TRANSPORTATION PROBLEM. It appears in many applications, for example where there is a group of sites which are candidates as plant locations and it is required to choose a subset of

these sites where plants should be located so as to minimize the total cost.

Due to the presence of the F_i 's and to the fact that supply would exceed demand if all factories are in operation, this problem has a different formulation than that of the classical transportation problem. It is formulated as to minimize:

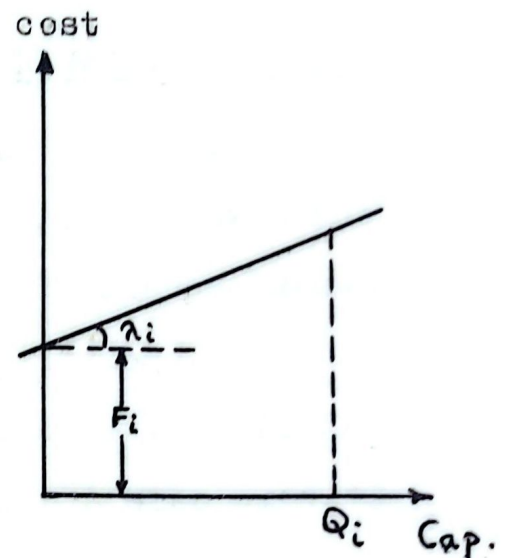
$$\Phi = \sum_i F_i y_i + \sum_i \sum_j (\lambda_i + c_{ij}) x_{ij}$$

$$\text{s.t.} \quad \sum_j x_{ij} \leq y_i Q_i$$

$$\sum_i x_{ij} \geq D_j$$

$$y_i \in \{0,1\} \text{ and } x_{ij} \geq 0$$

$$y_i \begin{cases} = 1 \rightarrow x_{ij} \geq 0 \quad \forall i,j \\ = 0 \rightarrow x_{ij} = 0 \quad \forall i,j \end{cases}$$



and the problem has to be solved as an INTEGER PROGRAM

REFERENCES

- 1) Hissch, W.M., and G.B.Dantzling, "The Fixed Charge Problem," The RAND Corporation, Paper P-648, Dec. 1954.

THE TRANSSHIPMENT PROBLEM

In some real world transportation studies the situation may arise whereby the commodity is subjected to **many** physical transportation processes, which need not be identical, en route to its final destinations. This is best illustrated by the bulk transportation of a manufactured commodity from the plants where it is manufactured to some intermediate warehouses from which it is transshipped to individual customers. Several remarks are in order:

- 1- The intermediate points (e.g. warehouses) act as destinations and sources simultaneously: they receive the commodity ~~from~~ the plants and send it to its final destinations.
- 2- The flow through each intermediate point varies according to the cost configuration for the entire system. However, this flow cannot exceed the upper bound on the intermediate point capacity.
- 3- The conservation of flow conditions state that the quantities shipped out of the intermediate point cannot exceed those shipped to it.
- 4- The cost/unit distance in and out of an inter-

mediate point could be different since the modes of transport could be different (bulk delivery vs van delivery).

- 5- Since there are so many sources at the two spatial stages, it may be more convenient to consider each slack capacity separately. This is achieved by adopting the notion that each source can ship to itself (at the appropriate underutilization penalty) as well as shipping to the destinations.
- 6- In general, there may be more than two spatial stages.
- 7- It **is possible** that the commodity receives some kind of processing at the intermediate points. If so, the resulting per unit cost is added to the shipping cost/unit for the outgoing traffic.

EXAMPLE: Consider a system incorporating 3 factories, 8 destinations and 4 depots. The information on availabilities, demands and depot capacities are given below and the transportation costs are given in tables 1 and 2

AVAILABILITIES	500	800	300		
DEPOTS CAPACITIES	600	1000	1000	300	
DEMANDS	{	180	230	200	220
		140	300	170	160

TABLE 1 : FACTORIES / DEPOTS TP COSTS

DEPOT FACTORY	I	II	III	IV
A	18	33	25	43
B	21	16	20	19
C	37	35	40	22

TABLE 2 : DEPOTS / DESTINATIONS TP COSTS

DESTINAT. DEPOT	1	2	3	4	5	6	7	8
I	13	31	27	17	29	19	21	23
II	5	22	14	33	25	16	26	14
III	26	28	18	30	8	24	15	32
IV	20	7	32	23	19	12	9	28

When this problem is formulated as a classical TP, the problem displayed in TABLE 3 is obtained.

TABLE 3 : TRANSSHIPMENT TABLE FOR THE 2-STAGE PROBLEM*

DEST. SOURCE	I	II	III	IV	1	2	3	4	5	6	7	8	AVAIL- ABILITY
A	500	18	33	25	43								500
B		21	800	16	19								800
C		37		300	22								300
I	100	0			180	15		220	17		100		600
II		200	0			31	100		29	19			
III			1000			22	14	33	25	16	70	160	1000
IV						26	18	30	8	24			1000
						100	100		19		100		300
DEMANDS	600	1000	1000	300	180	230	200	220	140	300	170	160	

(a)

(a)

* Since supply = demand, there is no need to make arrangements for slack capacities at the factories.

Once the TRANSSHIPMENT TABLEAU is obtained, the problem can be solved as a classical TP. However, it is possible to benefit from the transshipment feature in speeding up the calculations in a number of ways. Here we demonstrate the use of one feature is speeding up the MOD1 calculations in the following two processes:

A) Obtaining an Initial Feasible Solution:

The diagonal ad in table 3 has a unique property, it reflects the needs of the second stage in terms of the output from the first. For example, in the shown feasible flow, if depot I receives 500 units, we know that in the second stage, it is going to supply these 500 units to the destinations to conserve the material flow. Thus the REFLEXIVE DIAGONAL can be used in the process of entablising an initial feasible solution efficiently as follows:

- 1) Use any of the standard rules for the classical TP to find an initial feasible solution to the part of the tableau for the first stage only.
- 2) Having known the inputs to the various intermediate points, move to the lower part of the transshipment tableau and allocate the output from each intermediate point to the various destinations by any of the standard rules for the classical TP.

In conclusion, the property of reflexive diagonal allows us to find an initial solution by finding an initial feasible solution to two small classical TP's.

The same analysis applies to the cases where there are more than two stages.

B) Calculating the dual variables:

Each entry on the reflexive diagonal has two dual variables u_i, v_j related to it. If there is a basic variable in that cell, then we know that $c_{ij} = u_i + v_j$. However, it is a common practice to assume that all c_{ij} 's on the reflexive diagonal are zero.

If we agree to this and accept the assumption, then we have:

$$u_i = -v_j \quad \forall \{ (i,j) \text{ on a d} \mid x_{ij} > 0 \}$$

the procedure for calculating the dual variables can be augmented to utilize this property.

PRODUCTION AND DISTRIBUTION PROBLEMS:
AN APPLICATION OF TRANSSHIP-
MENT

Consider the following manufacturing and distribution system:

- Four factories A, B, C and D have variable production capacities.
- Each factory has an upper bound on its intake of raw material.
- Raw material can be processed into either:
 - a semifinished state (a),
 - or : a finished state (b),
- The semi-finished product (a) can be further processed (at any of the three factories) into 3 finished products a_2 , a_3 , a_4 .
- If the second process is carried out at a different factory, then the inter-factory cost is incurred.
- The finished products are distributed to 6 customers whose demands are known.
- Given the data shown in EXHIBIT I, decide upon the optimal production and distribution schedule.

EXHIBIT I: DATA FOR THE PRODUCTION/ DISTRIBUTION PROBLEM

UPPER BOUNDS ON THE PRODUCTION CAPACITIES

FACTORY	PRODUCT					TOTAL
	a	b	a ₂	a ₃	a ₄	
A	80	40	40	60	30	100
B	60	30	0	40	30	75
C	35	30	30	0	0	50
D	50	0	30	0	0	50

DEMANDS

DEST. PRODUCT	1	2	3	4	5	6
b	10	20	40	0	5	5
a ₂	10	5	0	25	15	15
a ₃	0	5	10	15	0	30
a ₄	5	10	0	0	20	25

DISTRIBUTION COSTS

	PRODUCT AND FACTORY										
	b			a ₂				a ₃		a ₄	
	A	B	C	A	B	C	D	A	B	A	B
1	3	8	4	6	16	7	12	11	27	10	18
2	2	5	2	3	9	4	4	5	15	3	10
3	3	4	1	5	7	3	5	8	12	6	8
4	7	1	5	12	1	10	11	21	2	14	1
5	6	2	5	12	3	9	10	20	5	13	4
6	6	4	6	11	7	10	7	18	12	12	8

EXHIBIT I (CONTINUED)

INTER. FACTORY TP COSTS

	A	B	C	D
A	0	7	3	3
B	7	0	6	6
C	3	6	0	4
D	3	6	4	0

Formulation of the Problem as a Transshipment Problem:

- Each factory can have up to 6 sources associated with it:
 - . 1 source representing the upper bound on the total capacity.
 - . 1 source " " " " " "
capacity for (a)
 - . 4 sources " " " bound for (b), a_2, a_3, a_4
- From each factory:
 - . The source representing total cap. can send to destinations for products (a), (b) at the same factory.
 - . Product (a) from one factory supplies destinations (a), a_2, a_3, a_4 at its own factory as well as at other factories.
 - . The appropriate cost estimates, capacity and demand figures are selected from tables in EXHIBIT I

The resulting tableau is as shown on the following page.

Solution for a production and distribution problem

		Factory 1			Factory 2			Factory 3			Factory 4			1						2		3		4		5		6		D U M M Y		A V A I L		
		A	B	A ₁	A ₂	A ₃	A	B	A ₁	A ₂	A ₃	A	B	A ₁	A ₂	A ₃	B	A ₁	A ₂	A ₃	B	A ₁	A ₂	A ₃	B	A ₁	A ₂	A ₃	B	A ₁	A ₂	A ₃		
E ₁		50	35																														100	
A		5		15	30	30	7																										80	
B			5																														40	
A ₁				25																													40	
A ₂					30																												60	
A ₃						30																											30	
E ₂							50	15																									75	
A								30	30																								60	
B								15																									30	
A ₁									10																								40	
A ₂																																	30	
A ₃																																	30	
E ₃										20	30																						50	
A											10																						35	
B																																	30	
A ₁																																	30	
A ₂																																	30	
A ₃																																	30	
E ₄																																	50	
A																																	50	
A ₁																																	30	
DEM		80	40	40	60	30	60	30	40	30	35	30	30	50	30	10	10	5	20	5	5	10	40	10	25	15	5	15	20	5	15	30	25	5

SITUATIONS WITH ALTERNATIVE METHODS:

AN APPLICATION OF TRANSSHIPMENT

Consider the situation of 3 factories A, B, C and 5 destinations 1, 2, 3, 4 & 5 and 2 methods of transportation: Road and Rail. Each location has its known capacities of loading and unloading bays for both methods. Also capacities of the factories are known together with demands at destinations and the TP costs. It may be required to decide upon the optimum shipping plan. In the present context we illustrate how this problem can be set up and solved as a transshipment problem.

The commodity will be assumed to flow in 3 stages:

- * Stage 1: From the factory to one of its loading bays (rail or road).
- * Stage 2: From the loading bay at the factory to the unloading bay at the destination.
- * Stage 3: From the unloading bay at the destination to the inside.

The cost elements for the 3 stages are as follows:

- * Stage 1: The loading costs at the factories.
- * Stage 2: The source/ destination cost (by the corresponding method).
- * Stage 3: The unloading costs at the destinations.

Notice the following:

- Each loading (unloading) point is a transshipment point. It may be convenient to assume the cost of its slack capacity equal to zero.
- The loading (unloading) capacities must be such as to allow for the existence of a feasible flow. For example, it could happen that the lack of unloading capacities at a certain destination results in the inability to meet its demand anyhow. A similar situation at a source could result in locking a part of its capacity.

EXAMPLE: Consider the data given in EXHIBIT II. The resulting transshipment tableau is shown after the exhibit.

EXERCISE: From the RAW DATA given in EXHIBITS I & II write the appropriate MG'S to solve the 2 problems once by the Revised Simplex and once by the classical TP method. Compare CPU times.

Notice that the MG for the TP formulation can be obtained from that for the LP by a TRIVIAL TRICK (??)

EXHIBIT II: DATA FOR THE ALTERNATIVE
METHODS EXAMPLE

CAPACITIES, DEMANDS, UPPER BOUNDS AND
LOADING / UNLOADING COSTS

LOCATION	A	B	C	1	2	3	4	5
QUANTITIES	5000	1000	1000	3000	3000	10000	5000	4000
U.B.'s { ROAD	3000	7000	5000	2000	1000	5000	2000	3000
RAIL	3000	4000	7000	2000	3000	6000	4000	2000
COSTS { ROAD	2	1	3	4	2	1	3	5
RAIL	2	5	3	1	4	2	5	3

TRANSPORTATION COSTS

	RAIL					ROAD				
	1	2	3	4	5	1	2	3	4	5
A	6	4	7	6	5	5	7	10	5	3
B	7	7	8	10	7	8	6	9	12	4
C	9	6	11	9	14	10	9	8	10	15

Transportation with different methods available

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1
--	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

REFERENCES

- 1 Orden, A., "The Transshipment Problem", Management Science,
Vol 2, 1956, pp. 276 - 285.

THE SOLID TRANSPORTATION PROBLEM

Consider the situation where there a "p" homogenous commodities produced at m sources and demanded at n destinations. Suppose that it is required to decide upon the shipping plan which minimizes the total cost subject to the additional requirement, over the ordinary supply/demand restrictions, that the total flow on route(i,j) is equal to a preset figure E_{ij} . In this case the problem is formulated as follows:

$$\text{Minimize } \phi = \sum_{k=1}^p \sum_{i=1}^n \sum_{j=1}^m C_{ijk} X_{ijk}$$

$$\text{Subject to } \sum_{k=1}^p$$

$$X_{ijk} = E_{ij} \quad \forall i, j \quad \text{DUAL } W_{ij}$$

$$\sum_{i=1}^n$$

$$X_{ijk} = D_{jk} \quad \forall j, k \quad U_{jk}$$

$$\sum_{j=1}^m$$

$$X_{ijk} = Q_{ki} \quad \forall k, i \quad V_{ki}$$

When

$$\sum_{i=1}^n \sum_{j=1}^m E_{ij} = \sum_{j=1}^m \sum_{k=1}^p D_{jk} = \sum_{k=1}^p \sum_{i=1}^m Q_{ki}$$

This structure of the problem and its treatment with a transportation type of approach, rather than a general LP, has been recognized by Haley^{*}. He termed it the Solid Transportation Problem (abbrev. STP in the sequel) because it can be visualized as a 3-dimensional block in which each layer corresponds to an ordinary transportation tableau for commodity k and the constraint $\sum_{k=1}^p X_{ijk} = E_{ij}$ (for all cells i,j) is the one that couples all the layers together. The solid structure is as displayed in Fig. 1

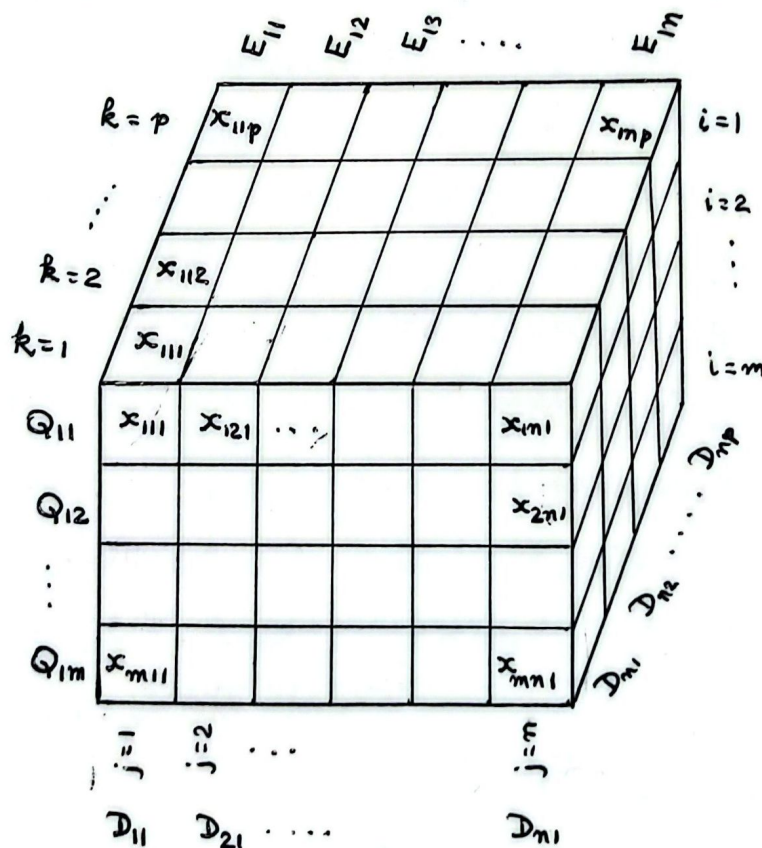


Fig. 1

^{*} Halay, K.B., "The solid Transportation Problem," Operations Research, Vol. 10, No. 4, 1962.

By analogy with the 2-dimensional TP (i.e. the classical TP), the following properties are obtained:

- 1) A basic feasible solution to the STP has $mnp - \{(m-1)(n-1)(p-1)\}$ variables in it. Notice that the classical TP method has $m \times n - \{(n-1)(m-1)\}$ variables in it.
- 2) When a basic feasible solution has been found, we can determine the dual variables by investigating the set of equations

$$W_{ij} + U_{ik} + V_{ki} = X_{ijk} \quad \forall \text{ basic variables}$$

However, we have $\{mnp - (m-1)(n-1)(p-1)\}$ such equations and $(m \times n) + (n \times p) + (p \times m)$ dual variables, so $(m+n+p-1)$ of these dual variables are given arbitrary values and we are left with K equations in K unknowns, where $K = mnp - \{(m-1)(n-1)(p-1)\}$. We return to the method of solving these equations at a later stage.

- 3) Once all the dual variables obtained, the feasibility of the dual program is checked. Consequently, we have to change the current primal solution if the following inequality holds for any non-basic cell:

$$W_{ij} + U_{ik} + V_{ki} > C_{ijk}$$

suppose that we choose $\hat{i} \hat{j} \hat{k}$ to enter the basis.

- 4) In relation to introducing $\hat{i} \hat{j} \hat{k}$, a basis loop has to be traced so as to recognize which basic variables are affected by this introduction and the quantity by which each is affected. Once these two targets are achieved, we alter the basic feasible solution accordingly and we go back to check its optimality.

Some Startling Difficulties In Handling the STP:

1) Conditions for the Existence of a Feasible Solution:

In the classical TP, the condition $\sum_{i=1}^m Q_i = \sum_{j=1}^n D_j$ is necessary and sufficient for the existence of a feasible solution to the problem. However, for the STP, the corresponding condition:

$$\sum_{i=1}^m \sum_{j=1}^n E_{ij} = \sum_{j=1}^m \sum_{k=1}^p D_{jk} = \sum_{k=1}^p \sum_{i=1}^m Q_{ki}$$

is necessary but NOT SUFFICIENT for the existence of a feasible solution. Up to the present day, to the best of my knowledge, the sufficient conditions are not yet known. However, it is an interesting research topic but most of the people keep away from it fearing a dead end.

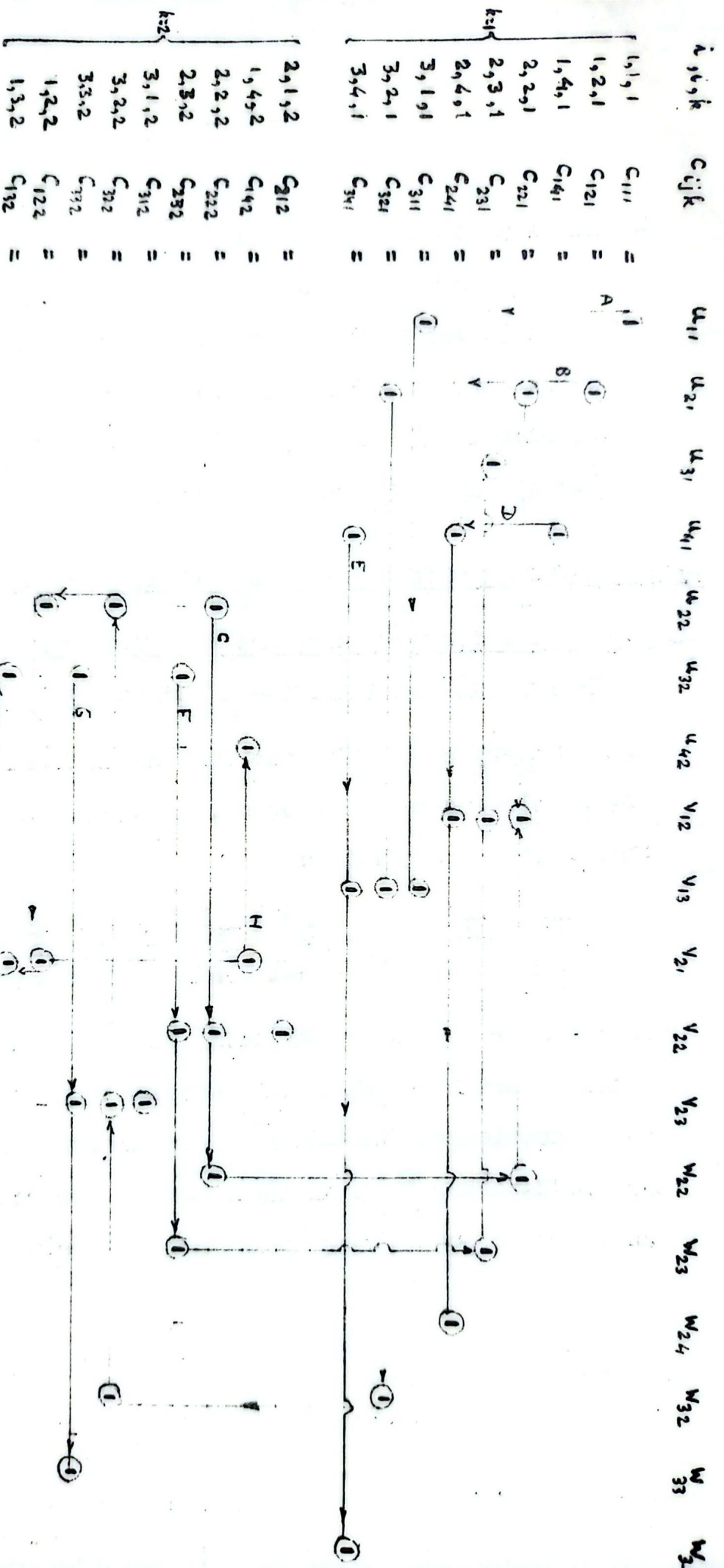


EXHIBIT I: DISPLAY OF THE MATRIX NEEDED TO DETERMINE DUAL

VARIABLES FOR THE GIVEN EXAMPLE. NOTICE

- Each letter Represents a Ra-start of the Scanning Routine
- Repetitive Scanning was able to calculate all dual cuts in this example. Generally, this is NOT THE CASE.

So unlike the classical TP, we cannot envisage the existence of a feasible solution to the STP before hand.

2) Calculating the dual variables:

According to the analysis, $(m+n+p-1)$ variables are chosen arbitrarily. However, one has to bear in mind that the variables have **to be selected** in such a way that the remaining variables form an independent set. The following rule was thought to be satisfactory:

- * Choose $W_{1j} = 0$ for $j=1, \dots, n$ (gives n variables)
- * Choose $W_{i1} = 0$ $i=2, \dots, m$ (gives $m-1$ variables)
- * Choose either U_{1k} or V_{k1} equal to zero for each layer k

(gives p variables) and this would result in a total of $(m+n+p-1)$ zero dual variables. Notice the resulting display for the given example with $p=2, m=3$ and $n=4$.

		U_{11}	U_{21}	U_{31}	U_{41}
$k=1$	$V_{11}=0$	4	1		9
	V_{12}		9	6	2
	V_{13}	4	1		6

		$U_{12}=0$	U_{22}	U_{32}	U_{42}
$K=2$	V_{21}		1	3	3
	V_{22}	3	3	3	
	V_{23}	4	4	1	

E_{ij} 's layer

0	0	0	0
0	W_{22}	W_{23}	W_{24}
0	W_{32}	W_{33}	W_{34}

Having chosen the arbitrary variables, we form the equation $W_{ij} + U_{jk} + V_{ki} = C_{ijk}$ corresponding to the basic variables. We notice that in general, each equation can contain more than 2 unknown dual variables and repetitive scanning is not guaranteed to lead to determining all the dual variables and we are generally left with a sub-system of linear equations in the remaining dual variables. This system has to be solved by one of the standard methods of linear algebra.

3) Recognizing the basis loop:

Because the basis in this problem is in 3-dimensions, the basis tree is also in 3-dimensions. The graph-theoretic approach which applies to the classical TP does not apply here at all, simply because we are dealing with 3-dimensional graphs. The properties of the basis loop are ill-defined and there is no way of stipulating its characteristics.

Any advancement in this direction is coupled with breakthroughs in 3-dimensional graph theory. There may be some time before any major breakthroughs in that direction could be expected. Actually, this is the main crossroads point at which the classical TP and the STP depart. One of them goes through an easy and through track, while the other, the STP, drives along a tedious (and may be blind) alley.

To the best of our knowledge, in order to determine the basis loop, we resort to basic principles of linear algebra and formulate the problem of tracing the basis loop as that of finding the alterations (μ_{ijk}) in the basic entries X_{ijk} so that the effect of entering the $(\hat{i}, \hat{j}, \hat{k})$ th variable in the basis balances out for each row, column in each layer,

$$\text{i.e.} \quad \sum_{k=1}^p \mu_{ijk} = 0$$

$$\sum_{i=1}^n \mu_{ijk} = 0$$

$$\sum_{j=1}^m \mu_{ijk} = 0$$

the only μ_{ijk} 's considered are those corresponding to the basic variables and that corresponding to $(\hat{i}, \hat{j}, \hat{k})$. However we know the $\mu_{\hat{i}\hat{j}\hat{k}} = 1$, so we can solve the residual system to get all the μ_{ijk} .

Each entry in the basis which has a non-zero μ_{ijk} is obviously an entry in the basis loop and is going to be affected by the quantity θ_{ijk} where θ is the value of the entering variable. Now we move on to the choice of θ .

To comply with primal feasibility, θ has to be chosen in such a manner that when the alterations $\theta \mu_{ijk}$ are performed on the entries of the basis loop, none of them becomes negative. Thus θ is chosen as:

$$\theta = \min \frac{X_{ijk}}{\mu_{ijk}} \quad \{ \text{s.t. } \mu_{ijk} \text{ is -ve} \},$$

Having known all the μ_{ijk} 's and the θ , we can update the variables in the basic solution. The new variables, together with the variable $X_{\hat{i}\hat{j}\hat{k}} = \theta$, we get the new basic feasible solution. However, few remarks are in order at this stage:

- a) The μ_{ijk} 's are not guaranteed to have integer values resulting from solving the system of linear equations. Consequently θ can have non-integer values and $\underline{x_{ijk}}$ can be, in general, non-integer. Note that this is a major difference between the STP and the classical TP.
- b) The above rule for featuring the basis loop also applies to the classical TP. In fact, this produces a very efficient basis loop tracing routine. In this case, we drop the index k and the above equations reduce to:

$$\left. \begin{array}{l} \sum_{i=1}^m \mu_{ij} = 0 \\ \sum_{j=1}^n \mu_{ij} = 0 \end{array} \right\} \forall (i,j) \text{ in the basis}$$

and $\mu_{ij} = 1$. Because of basis triangularity, the above system can be solved by repetitive scanning. Since $\mu_{ij} = 1$, all the other μ_{ij} 's will be integer and in fact, will have the values 1, 0, -1. Consequently $\theta = \min \frac{x_{ij}}{\mu_{ij}}$ (s.t. $\{\mu_{ij} = -1\}$). Notice that this is exactly what we do in any basis trace routine!!. Let us consider the following hand example for demonstration:

18	-1					18
4	+1	9	0	6	-1	19
				11	+1	21
					10	-1

		22	9	17	10		
μ_{11}	μ_{21}	μ_{22}	μ_{23}	μ_{33}	μ_{34}	μ_{14}	
1						1	=0
1	1	1					=0
				1	1		=0
1	1						=0
		1					=0
			1	1			=0
					1	1	=0

It is a trivial matter to see that repetitive scanning results in the following values of the μ 's:

$$\mu_{22} = 0, \mu_{11} = -1, \mu_{21} = +1, \mu_{34} = -1, \mu_{33} = +1, \mu_{23} = -1$$

$$\theta = \min \{18, 6, 10\} = 6$$

Personally, I think this can produce the most efficient basis trace routine and I am quite willing to help somebody test this cinjecture as I never had the time to do it myself.

The STP in Prospective:

It remains to be an UNSOLVED and certainly a very challenging problem. Undoubtedly, it has a voluminous array of applications but people tend to keep quiet about them to hide the defficiency that this problem is still unsolved. We will continue with some types of its applications in a future context.

APPLICATIONS OF THE STP

The formulation of distribution problems as STP has many potential applications. However, this fact is overlooked in many situations due to the truth that the STP is still unsolved. In the present context we give **two** illustrations of the applications:

A) TRANSPORTATION USING DIFFERENT METHODS:

In a previous context we considered the situation where shipments from 3 factories to 5 destinations were carried by two modes of transport: rail and road. Due to insufficient capacities, both modes (rather than the cheapest between them) had to be used and the objective was to find the shipping plan which utilized both modes at the minimum overall cost.

This problem can be formulated as a STP as follows:

- 1- Each mode has its own layer **k** in the solid problem.
- 2- Row and column sums for each table are the limits on the amount that can be sent (received) by that factory (customer) via the corresponding mode of transport.
- 3- One extra row and column is introduced into each table to take care of the slacks.

- 4- The sum of the **slacks** for each factory is equal to the difference between the total loading capacity and the availability.
- 5- The sum of the **slacks** for each destination is equal to the difference between the total unloading capacity and the requirements.
- 6- The limit **on** shipments between factories and customers is unlimited and is conveniently represented by a large number R.

Thus the problem takes the form displayed in tables 1 → 4 in Exhibit I.

Notice that this structure imposes the negative costs for the **slack capacities**, thus the total cost is actually the cost obtained from the solution plus the terms representing the loading(unloading) capacities multiplied by the loading(unloading) costs.

This structure was imposed by the excess capacity feature in the original problem.

A feasible solution is shown in EXHIBIT II. Notice that it is the same as that shown before for the same problem.

EXHIBIT I: THE TWO-MODE TRANSPORTATION PROBLEM

RAIL	I	II	III	IV	V	SLACK	TOTAL
A	6	4	7	6	2	-2	3
B	7	7	8	10	7	-2	4
C	9	6	11	9	14	-3	7
SLACK	-1	-4	-2	-5	-3	0	17
TOTAL	2	3	6	4	2	14	

ROAD	I	II	III	IV	V	SLACK	TOTAL
A	5	7	10	5	3	-2	3
B	8	6	9	12	4	-1	7
C	10	9	8	10	15	-3	5
SLACK	-4	-2	-1	-3	-5	0	13
TOTAL	2	1	5	2	3	15	

SLACK	I	II	III	IV	V	SLACK	TOTAL
A	0	0	0	0	0	K	5R-5
B	0	0	0	0	0	K	5R-10
C	0	0	0	0	0	K	5R-10
SLACK	K	K	K	K	K	0	R
TOTAL	3R-3	3R-3	3R-10	3R-5	3R-4	R	

EXHIBIT I : (CONTINUED)

E_{ij} 's	I	II	III	IV	V	SLACK
A	R	R	R	R	R	$3+3$ -5
B	R	R	R	R	R	$7+4$ -10
C	R	R	R	R	R	$5+7$ -10
SLACK	$2+2$ -3	$1+3$ -3	$5+6$ -10	$2+4$ -5	$3+2$ -4	$R+25$

EXHIBIT II: SOLUTION OF THE 2-MODE STP

RAIL	I	II	III	IV	V	SLACK	TOTAL
A	2		1				3
B			4			8	4
C	8	2		3	2		7
SLACK		1	1	1		14	17
TOTAL	2	3	6	4	2	14	

ROAD	I	II	III	IV	V	SLACK	TOTAL
A	8			2		1	3
B	1	1	2		2	1	7
C			3			2	5
SLACK	1				1	11	13
TOTAL	2	1	5	2	3	15	

SLACK	I	II	III	IV	V	SLACK	TOTAL
A	R-2	R	R	R-3	R	8	5R-5
B	R-1	R-1	R-6	R	R-2		5R-10
C	R	R-2	R-4	R-2	R-2		5R-10
SLACK						R	R
TOTAL	3R-3	3R-3	3R-10	3R-5	3R-4	R	

B) THE CAPACITATED TRANSPORTATION PROBLEM:

This problem can be formulated as follows:

$$\text{Minimize } \phi = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij}$$

$$\text{Such that } \sum_{i=1}^m X_{ij} = D_j$$

$$\sum_{j=1}^n X_{ij} = Q_i$$

$$0 \leq X_{ij} \leq E_{ij}$$

To formulate this problem as a STP a problem with **two** layers is envisaged. The first layer houses the original problem and the second layer handles the slacks. Thus a new class of variables X_{ijk} is defined as follows: $\sum_{k=1}^n X_{ijk} = E_{ij}$ and the above equations are rewritten as:

$$\sum_{i=1}^m X_{ij1} = D_j$$

$$\sum_{j=1}^n X_{ij1} = Q_i$$

Thus the following is true

$$\sum_{i=1}^m X_{ij2} = \sum_{i=1}^m E_{ij} - \sum_{i=1}^m X_{ij1}$$

$$\text{i.e.,} \quad = \sum_{i=1}^m E_{ij} - D_j$$

$$\text{and} \quad \sum_{j=1}^n X_{ij2} = \sum_{j=1}^n E_{ij} - \sum_{j=1}^n X_{ij1}$$

$$= \sum_{j=1}^n E_{ij} - Q_i$$

So the STP reads as follows:

$$\text{Minimize} \quad \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^2 t_{ijk} X_{ijk}$$

$$\text{Subject to} \quad \sum_{j=1}^n X_{ijk} = A_{ki}$$

$$\sum_{i=1}^m X_{ijk} = B_{jk}$$

$$\sum_{k=1}^n X_{ijk} = E_{ij}$$

$$X_{ijk} \geq 0$$

$$\text{Where: } A_{1i} = Q_i \text{ and } A_{2i} = \sum_{j=1}^n E_{ij} - Q_i$$

$$\text{and } B_{j1} = D_j \text{ and } B_{j2} = \sum_{i=1}^m E_{ij} - D_j$$

$$\text{and } t_{ij1} = C_{ij} \forall i, j \text{ and } t_{ij2} = 0 \forall i, j$$

Exercise: We have discussed a capacitated TP problem in a previous lecture. Formulate that problem as a STP.

REFERENCES

- 1) Haley, K.B., "The Solid Transportation Problem," Operations Research, Vol. 10, 1962, pp. 448-463.
- 2) Haley, K.B., "The Multi-Index Problem," Operations Research, Vol. 11, 1963, pp. 368-379.

A Primal Out-of-Kilter Algorithm
For the Transportation Problem

Based on the earlier work of Balinski and Gomory (B-G) [1], Graciano Sà developed a "variant" which experimentally proved to be 20% faster [4]. His main contribution was that of ignoring the concept of maintaining orthogonality within each stage. He also allowed the algorithm to throw off-Kilter an in-Kilter activity at any stage of the calculation. This concept, usually termed "overshooting", was first introduced by Jewell [3] when he generalized the out-of-Kilter method to linear programs; but it was not generally practiced in previous out-of-Kilter methods for the transportation problem. Sà introduced some concepts and invoked notions which, we feel, are necessary to grasp before the general steps are outlined; namely:

a) Breakthrough^{*}:

If a unique sequence of cells which may trade off flow, without disturbing primal feasibility, is identified in the labelling process, the sequence forms a loop starting and ending in one cell and containing an even number of cells.

b) Nonbreakthrough:

is the case where the labelling procedure is unable to label the "starred" row or column from which labelling started.

c) Class A Cells:

are the cells representing infeasible (non-conforming, out-of-kilter) states. One attempts a corrective move by

* The concept of Breakthrough and Nonbreakthrough was first introduced by Fukerson [2].

adding flows. Such cells are given by:

$$(i,k) = \{(i,k) \mid t_{ik} < u_i + v_k \text{ and } x_{ik} < 0\}$$

d) Class B Cells:

are the cells representing non-orthogonal states. One attempts to correct (conform, throw in kilter) the flow by withdrawing flows. Such cells are given by:

$$(i,k) = \{(i,k) \mid t_{ik} > u_i + v_k \text{ and } x_{ik} > 0\}$$

e) Class A (Class B) Loops:

If the first cell (k, ℓ) in the loop is a class A cell (class B cell) we call the loop a class A loop (class B loop).

f) The sets J_1, J_2 & J'_1, J'_2 :

the following sets will be frequently formed and searched during some stages of the solution:

$$\begin{aligned} J_1 &= \{(i,j) \mid i \text{ labelled and } j \text{ unlabelled}\} \\ J_2 &= \{(i,j) \mid i \text{ unlabelled and } j \text{ labelled}\} \\ J'_1 &= \{(i,j) \mid (i,j) \in J_1 \text{ and } t_{ij} > u_i + v_j\} \\ J'_2 &= \{(i,j) \mid (i,j) \in J_2 \text{ and } t_{ij} < u_i + v_j\} \end{aligned}$$

6.1. General Steps of Sâ's Primal Algorithm

Step 1. Start with a dual pair $(X), (U,V)$ such that (X) is feasible and the pair satisfies the following condition:

$$Z = \sum_i \sum_k (t_{ik} - u_i - v_k) x_{ik} = 0 \quad (1)$$

Step 2. Check whether $t_{ik} - u_i - v_k \geq 0$ for all cells; if so go to step 19.

Otherwise there exists a column ℓ with out-of-kilter entries.

Step 3. At any general stage ℓ , Select (k, ℓ) such that:

$$u_k + v - t_k = \max_i (u_i + v_k - t_{ik}) = q > 0 \quad (2)$$

and concentrate on column ℓ first till all its cells become conforming.

Step 4. Star row k and assign label k to column ℓ

Step 5. Use the following set of rules for labelling

(L1) If column j has been labelled, label with j all unlabelled rows i for which

$$x_{ij} > 0 \text{ and } t_{ij} \geq u_i + v_j \quad (3)$$

(L2) If row i is labelled, label with i all unlabelled columns j for which

$$t_{ij} = u_i + v_j \quad (4)$$

Cycle through the two rules till no more labelling is possible; either of the following two cases arises:

(a) The row k receives a label; in this case continue to Step 6

(b) The row k does not receive a label; in this case go to Step 16

Step 6. There is a breakthrough and a loop can be uniquely

defined. Calculate

$$S = \min \left\{ x_{ij} \left| \begin{array}{l} (i,j) \text{ is even entry in a class A loop} \\ \text{or odd entry in a class B loop} \end{array} \right. \right\} \quad (5)$$

Step 7. Update the flow as follows $\star$:

$$\left. \begin{array}{ll} \text{Class A loops: } x_{ij} := x_{ij} + S & (i,j) \text{ odd} \\ & x_{ij} := x_{ij} - S & (i,j) \text{ even} \\ \text{Class B loops: } x_{ij} := x_{ij} - S & (i,j) \text{ odd} \\ & x_{ij} := x_{ij} + S & (i,j) \text{ even} \end{array} \right\} \quad (6)$$

Step 8. Update the dual variable of column l and the costs as follows: $v_l := v_l - q$ (7)

$$(t_{il} - u_i - v_l) := (t_{il} - u_i - v_l) + q \quad \forall i \quad (8)$$

Step 9. Check whether $S = S_l$; if so all entries in l are made conforming when updated as in Step 8 and the stage is finished; return to Step 2. Otherwise

Step 10. If $S < S_l$, this means that (8) "overshoots" and at least one out-of-kilter cell (t, l) such that

$$x_{tl} > 0 \text{ and } (t_{tl} - u_t - v_l) = q > 0$$

is introduced in column l . Cell (t, l) is a

class B cell, so "star" column ℓ and assign label ℓ to row t .

Step 11. Drop the existing labels and cycle through rules (L1) and (L2) till no further labelling is possible. Either of the following two cases arises:

- (c) The column ℓ receives a label; in this case continue to Step 12.
- (d) The column ℓ does not receive a label; go to Step 16

Step 12. There is a breakthrough,

- Calculate S as in (6)
- Update flows as in (7)

Step 13. Check whether $S = x_{t\ell}$; if so cell(t, ℓ) is made conforming with the flow change. In any case proceed to the next step:

Step 14. Scan column ℓ for remaining class-B-cells. If none exists the stage is closed; return to Step 2. Otherwise.

Step 15. If some cells of class B still exist, return to Step 10

Step 16. If there is a Nonbreakthrough (the starred row or column is not labelled) define:

$$\psi = \begin{cases} \min_{j_1} (t_{ij} - u_i - v_j) > 0 & \text{if } J_1' = \emptyset \\ - (t_{ij} - u_i - v_j) & \text{if } J_1' \neq \emptyset \end{cases} \quad (9)$$

Step 17. Update the dual vectors as follows:

$$\begin{aligned} u_i &:= u_i - \psi & \text{if } i \text{ is not labelled} \\ v_j &:= v_j + \psi & \text{if } j \text{ is not labelled} \end{aligned} \quad (10)$$

Update the costs as follows:

$$\left. \begin{aligned} (t_{ij} - u_i - v_j) &:= (t_{ij} - u_i - v_j) - \psi & (i,j) \in J_1 \\ (t_{ij} - u_i - v_j) &:= (t_{ij} - u_i - v_j) + \psi & (i,j) \in J_2 \end{aligned} \right\} (11)$$

Step 18. If, upon updating a class A cell (k, ℓ) is still non-conforming, go to Step 3; Otherwise go to Step 5.

Step 19. Terminate

REFERENCES

- 1) Balinski, M.L. and R.E. Gomory, "A Primal Method for the Assignment and Transportation Problems, Management Science, Vol 10, No. 4, 1964.
- 2) Fulkerson, D.R., "An Out-of-Kilter Method for Minimal Cost Flow Problems," Journal of SIAM, Vol. 9, No. 1, 1961.
- 3) Jewell, W.S., "COMPLEX: A Complementary-Slackness Out-of-Kilter Algorithm for Linear Programming, "Report ORC 67-6, Operations Research Center, University of California, Berkley, March 1967.

- (4) Sa, G., "Concave Programming in Transportation Networks,
"Ph.D. Thesis, MIT School of Management, Sept, 1968,.

INSTRUCTIONS FOR USING GRACE
A PRIMAL OUT-OF-KILTER TRANSPORTATION CODE

In order to initiate the code, you need to initiate it with a primal feasible solution (obtained by HADY for example). Both the X matrix and the U,V vectors are needed together with an S matrix defined as follows:

$$s_{ij} = \begin{cases} 1 & \text{if } x_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The parameters and arrays are displayed as follows:

M: Number of sources
N: Number of destination
K(I): A vailability at source i
R(J): Demand at destination j
C(I,J): TP cost for route i,j
KEY : $\begin{cases} 1 & \text{if the operation with GRACE is successful} \\ -1 & \text{otherwise} \end{cases}$

and is returned via the subroutine argument to the calling segment. It has to be initiated = 1 before GRACE is called in the first place.


```

C *****
C   A SUBROUTINE FOR THE SOLUTION OF THE TRANSPORTATION PROBLEM
C   USING A MODIFIED VERSION OF THE BALINSKI-GOMORY'S OUT-OF-KILTER
C   ALGORITHM
C -----
C   SUBROUTINE GRACE(KEY)
C     INTEGER STAGE,AUX,STEP2
C     COMMON U(10),V(10),X(10,10),K(10),R(10),C(10,10),S(10,10)
C     COMMON IIBAS(20),IJBAS(20),IVBAS(20)
C     COMMON LU(10),LV(10)
C     COMMON M,N,IND1,IND,NN
C     STAGE=0
10  DO 70 I=1,M
    IF(K(I))30,70,30
30  DO 50 J=1,N
    IF(S(I,J))40,40,50
40  X(I,J)=C(I,J)-U(I)-V(J)
50  CONTINUE
70  CONTINUE

C     WRITE(3,6)

C     CHECK WHETHER THE SOLUTION AT HAND IS OPTIMAL OR ANEW STAGE IS DUE
C
90  MIN3=0
    MIN2=0
    DO 300 I=1,M
    IF(K(I))110,300,110
110  DO 280 J=1,N
    IF(S(I,J))130,130,170
130  IF(MIN3-X(I,J))260,280,150
150  MIN3=X(I,J)
    KK=I
    LI=J
    GO TO 280
170  AUX=C(I,J)-U(I)-V(J)
    IF(AUX)200,280,250
200  V(J)=V(J)+AUX
    DO 220 II=1,M
    IF(S(II,J))210,210,220
210  X(II,J)=X(II,J)-AUX
220  CONTINUE
    GO TO 90
250  JE=AUX*X(I,J)
    IF(MIN2-JE)260,280,280
260  MIN2=JE
    KK1=I
    LL1=J
280  CONTINUE
300  CONTINUE
    WRITE(3,11)

C     IF(MIN3)320,310,320
310  IF(MIN2)320,1460,320
320  IF(MIN3)350,370,370
350  STLP2=0

```

```
GO TO 390
370 STEP2=1
KK=KK1
L1=LL1
390 STAGE=STAGE&1
ICL1=R(L1)
LMSTP=M*ICL1
IF(N-STAGE)410,430,430
C
410 WRITE(3,3)STAGE,N
C
KEY=-1
C
RETURN
C
C
430 NSTEP=0
C
C BEGIN A FRESH STAGE, NO HANGOVERS, DROP ALL LABELS
450 JT=1
WRITE(3,12)
DO 470 I=1,M
470 LU(I)=0
DO 490 J=1,N
490 LV(J)=0
NSTEP=NSTEP&1
IF(LMSTP-NSTEP)510,530,530
510 WRITE(3,4)NSTEP,L1,LMSTP
C
KEY=-1
C
RETURN
C
530 IF(STEP2)560,560,540
540 LU(KK)=L1
GO TO 690
C ASSIGN LABEL KK TO COLUMN L1
560 LV(L1)=KK
C
C START LABELLING
C
C
580 IE=1
WRITE(3,13)
C IF COLUMN J HAS BEEN LABELLED, THEN LABEL WITH(J) ALL ROWS I HAVING
C X(I,J).GT.0 WHICH HAVE NOT YET BEEN LABELLED
DO 670 J=1,N
IF(LV(J))600,670,600
600 IF(STEP2)610,610,605
605 IF(J-L1)610,1300,610
610 DO 650 I=1,M
C
IF(S(I,J))650,650,620
620 IF(LU(I))630,630,650
630 IF(K(I))640,650,640
IE=0
```

```
640 LU(I)=J
650 CONTINUE
C
670 CONTINUE
C
      IF(IE)690,690,800
C      IF ROW I (I.NE.KK) HAS BEEN LABELLED, THEN LABEL WITH(I) ALL COLUM
C      *NS J HAVING U(I)&V(J)=C(I,J) WHICH HAVE NOT YET BEEN LABELLED
690 IF=0
      WRITE(3,14)
      DO 790 I=1,M
C
      IF(LU(I))710,790,710
710 IF(K(I))720,790,720
720 IF(I-KK)745,735,745
735 IF(STEP2)1240,1240,745
C
745 DO 780 J=1,N
      IF(C(I,J)-(U(I)&V(J)))780,750,780
750 IF(LV(J))760,760,780
760 LV(J)=I
      IE=1
C
780 CONTINUE
790 CONTINUE
C
      IF(IE)800,800,580
C
C      SORRY, NO BREAKTHROUGH, ROW KK IS NOT LABELLED, CASE 3 OCCURS
C
800 IM=32767
      WRITE(3,16)
      DO 880 I=1,M
      IF(LU(I))810,880,810
810 IF(K(I))820,880,820
820 DO 860 J=1,N
      IF(LV(J))840,840,860
840 AUX=C(I,J)-U(I)-V(J)
      IF(AUX)860,860,845
845 IF(IM-AUX)860,860,850
850 IM=AUX
860 CONTINUE
C
880 CONTINUE
C
      IF(IM-32767)900,890,900
890 IM=-X(KK,11)
C
C      UPDATE THE U'S
C      WRITE(3,17)
900 DO 970 I=1,M
      IF(K(I))920,970,920
920 IF(LU(I))930,930,940
930 U(I)=U(I)-IM
      GO TO 970
C
```



```

C      UPDATE THE X(I,J)'S FOR NONBASIC ENTRIES
940 DO 960 J=1,N
      IF(S(I,J))945,945,960
945 IF(LV(J))950,950,960
950 X(I,J)=X(I,J)-IM
960 CONTINUE
970 CONTINUE

C
C      UPDATE THE V'S
      WRITE(3,18)
      DO 1030 J=1,N

C
      IF(LV(J))990,990,1000
990 V(J)=V(J)&IM
      GO TO 1030

C
C      UPDATE THE X(I,J)'S FOR NONBASIC ENTRIES
1000 DO 1025 I=1,M
      IF(S(I,J))1005,1005,1025
1005 IF(LU(I))1010,1010,1025
1010 IF(K(I))1020,1025,1020
1020 X(I,J)=X(I,J)&IM
1025 CONTINUE

C
1030 CONTINUE

C
C      CHECK IF COLUMN L1 IS ELIGIBLE TO START ANEW STEP
1050 MIN3=0
      WRITE(3,19)
      MIN2=0
      IF(S(KK,L1))1060,1060,1055
1055 IF(C(KK,L1)-(U(KK)&V(L1)))1060,1060,1210
1060 DO 1150 I=1,M
      IF(K(I))1090,1150,1090
1090 IF(S(I,L1))1100,1100,1120
1100 IF(X(I,L1)-MIN3)1110,1150,1150
1110 MIN3=X(I,L1)
      KK2=I
      WRITE(3,21)
      GO TO 1150
1120 AUX=C(I,L1)-U(I)-V(L1)
      IF(MIN2-(AUX*X(I,L1)))1140,1150,1150
1140 MIN2=AUX*X(I,L1)
      KK1=I

C
1150 CONTINUE

C
      IF(MIN3)1170,1190,1190
1170 IF(KK2-KK)1180,1210,1180
1180 KK=KK2
      WRITE(3,22)

C
      GO TO 450

C
1190 IF(MIN2)90,90,1200
C

```

```
1200 STEP2=1
      KK=KK1
C      START A NEW STAGE, DROP LABELS
C
      WRITE(3,26)
      GO TO 450
C
C      COLUMN L1 PASSED TEST
1210 IF(JT)450,450,1220
1220 NSTEP=NSTEP&1
      WRITE(3,27)
C      START ANEW STEP, MAINTAIN LABELLS
      GO TO 690
C
C      BREAKTHROUGH, ROW KK RECEIVES A LABEL, A LOOP IS UNIQUELY DEFINED
1240 IM=-(C(KK,L1)-U(KK)-V(L1))
      WRITE(3,28)
      V(L1)=V(L1)-IM
      DO 1280 I=1,M
      IF(K(I))1260,1280,1260
1260 IF(S(I,L1))1270,1270,1280
1270 X(I,L1)=X(I,L1)&IM
1280 CONTINUE
C
1300 I1=KK
      IM=32767
      WRITE(3,29)
      GO TO 1360
C
C      EVALUATE THE FLOW CHANGE
C
1330 I1=LV(J1)
      IF(I1-KK)1360,1340,1360
1340 IF(STEP2)1350,1350,1410
1350 IF(J1-L1)1360,1410,1360
1360 J1=LU(I1)
      IF(IM-X(I1,J1))1330,1330,1370
1370 IM=X(I1,J1)
      WRITE(3,31)
      GO TO 1330
C
C      PERFORM THE FLOW CHANGE
1400 I1=LV(J1)
      WRITE(3,32)
      X(I1,J1)=X(I1,J1)&IM
      S(I1,J1)=1
      IF(I1-KK)1410,1402,1410
1402 IF(STEP2)1405,1405,1430
1405 IF(J1-L1)1410,1430,1410
C
1410 J1=LU(I1)
      IF(X(I1,J1))1420,1420,1400
1420 S(I1,J1)=0
      X(I1,J1)=C(I1,J1)-U(I1)-V(J1)
      WRITE(3,33)
      GO TO 1400
```

```
C
C      UPDATE DEMAND FOR COLUMN L1 AND PRINT TABLE
1430 ICL1=ICL1-IM
C
      DO 1440 I=1,M
1440 WRITE(3,2)(X(I,J),J=1,N)
      WRITE(3,7)
C
      DECIDE WHETHER TO START ANOTHER STEP OR A NEW STAGE
      JT=0
      IF(ICL1)90,90,1050
C
      FINAL CHECK BEFORE TERMINATION
1460 DO 1500 I=1,M
      DO 1500 J=1,N
      IF(K(I))1465,1500,1465
1465 IF(S(I,J))1480,1480,1470
1470 IF(C(I,J)-(U(I)&V(J)))1480,1500,1480
1480 IF(S(I,J))1485,1485,1490
1485 IF(X(I,J))1490,1500,1500
1490 WRITE(3,9)I,J
      KEY=-1
C
      RETURN
C
1500 CONTINUE
C
      RETURN
C
      2 FORMAT(10X,16F6.0)
      3 FORMAT(10X,'NUMBER OF STAGES',I5,'EXCEEDS BOUND',I5)
      4 FORMAT(10X,'NUMBER OF STEPS ',I5,'IN COLUMN',I5,'EXCEEDS BOUND',I5)
      6 FORMAT(1H1)
      7 FORMAT(//)
      9 FORMAT(10X,'CELL I ='I3,' J ='I3,'NON CONF')
C
11 FORMAT(10X,'I HAVE CHECKED INITIAL SOLUTION')
12 FORMAT(10X,'I AM BEGINING ANEW STAGE AT450')
13 FORMAT(10X,'I WILL TRY TO LABEL ROWS AT 580')
14 FORMAT(10X,'I WILL TRY TO LABEL COLUMNS AT 690')
16 FORMAT(10X,'I AM NOW AT 800')
17 FORMAT(10X,'I HAVE JUST FINISHED 890')
18 FORMAT(10X,'HAVING FINISHED 970 I WILL UPDATE THE V S')
19 FORMAT(10X,'BEING AT 1050 IWILL CHECK WHETHER L1 WILL STEP')
21 FORMAT(10X,'STILL WITH L1 ARE GOING TO 1150')
22 FORMAT(10X,'L1 IS NO GOOD I WILL GO TO 450 AND START STAGE')
26 FORMAT(10X,'I WILL CHANGE MY LABELLS AND GO TO 450 ')
27 FORMAT(10X,'WHAT A NASTY COLUMN L1 ISWF KEEP LABELLS AND TO 690')
28 FORMAT(10X,'KK HAD ALABELL HA HA')
29 FORMAT(10X,'I AM NOW AT 1300')
31 FORMAT(10X,'AT LAST I AM ABLE TO EVALUATE A FLOW CHANGE')
32 FORMAT(10X,'AND OFF WE GO CHANGE FLOW PEU INEED A REST')
33 FORMAT(10X,' STILL LIMPING I AM GOING TO 1400')
C
      END
```