H03ABF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

H03ABF solves the classical Transportation ('Hitchcock') problem.

2 Specification

```
 SUBROUTINE H03ABF(KOST, MMM, MA, MB, M, K15, MAXIT, K7, K9, NUMIT,
1                  K6, K8, K11, K12, Z, IFAIL)
 INTEGER           KOST(MMM,MB), MMM, MA, MB, M, K15(M), MAXIT,
1                  K7(M), K9(M), NUMIT, K6(M), K8(M), K11(M),
2                  K12(M), IFAIL
 real              Z
```

3 Description

H03ABF solves the Transportation problem by minimizing

$$z = \sum_i \sum_j c_{ij} x_{ij}.$$

subject to the constraints

$$\sum_j x_{ij} = A_i \text{ (Availabilities)}$$

$$\sum_i x_{ij} = B_j \text{ (Requirements)}$$

where the x_{ij} can be interpreted as quantities of goods sent from source i to destination j, for $i = 1, 2, \ldots, m_a$; $j = 1, 2, \ldots, m_b$, at a cost of c_{ij} per unit, and it is assumed that $\sum\limits_i A_i = \sum\limits_j B_j$ and $x_{ij} \geq 0$.

H03ABF uses the 'stepping stone' method, modified to accept degenerate cases.

4 References

[1] Hadley G (1962) *Linear Programming* Addison–Wesley

5 Parameters

1: KOST(MMM,MB) — INTEGER array *Input*

On entry: the coefficients c_{ij}, for $i = 1, 2, \ldots, m_a$; $j = 1, 2, \ldots, m_b$.

2: MMM — INTEGER *Input*

On entry: the first dimension of the array KOST as declared in the (sub)program from which H03ABF is called.

Constraint: MMM $\geq$ MA.

3: MA — INTEGER *Input*

On entry: the number of sources, m_a.

Constraint: MA ≥ 1.

4: MB — INTEGER *Input*

On entry: the number of destinations, m_b.

Constraint: MB ≥ 1.

5: M — INTEGER *Input*

On entry: the value of $m_a + m_b$.

6: K15(M) — INTEGER array *Input/Output*

On entry: K15(i) must be set to the availabilities A_i, for $i = 1, 2, \ldots, m_a$; and K15($m_a + j$) must be set to the requirements B_j, for $j = 1, 2, \ldots, m_b$.

On exit: the contents of K15 are undefined.

7: MAXIT — INTEGER *Input*

On entry: the maximum number of iterations allowed.

Constraint: MAXIT ≥ 1.

8: K7(M) — INTEGER array *Workspace*

9: K9(M) — INTEGER array *Workspace*

10: NUMIT — INTEGER *Output*

On exit: the number of iterations performed.

11: K6(M) — INTEGER array *Output*

On exit: K6(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the source indices of the optimal solution (see K11 below).

12: K8(M) — INTEGER array *Output*

On exit: K8(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the destination indices of the optimal solution (see K11 below).

13: K11(M) — INTEGER array *Output*

On exit: K11(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the optimal quantities x_{ij} which, sent from source i = K6(k) to destination j = K8(k), minimize z.

14: K12(M) — INTEGER array *Output*

On exit: K12(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the unit cost c_{ij} associated with the route from source i = K6(k) to destination j = K8(k).

15: Z — ***real*** *Output*

On exit: the value of the minimized total cost.

16: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, the sum of the availabilities does not equal the sum of the requirements.

IFAIL = 2

During computation MAXIT has been exceeded.

IFAIL = 3

On entry, MAXIT < 1.

IFAIL = 4

On entry, MA < 1,
or MB < 1,
or M $\neq$ MA + MB,
or MA > MMM.

7 Accuracy

All operations are performed in integer arithmetic so that there are no rounding errors.

8 Further Comments

An accurate estimate of the run time for a particular problem is difficult to achieve.

9 Example

A company has three warehouses and three stores. The warehouses have a surplus of 12 units of a given commodity divided among them as follows:

Warehouse	Surplus
1	1
2	5
3	6

The stores altogether need 12 units of commodity, with the following requirements:

Store	Requirement
1	4
2	4
3	4

Costs of shipping one unit of the commodity from warehouse i to store j are displayed in the following matrix:

		Store		
		1	2	3
	1	8	8	11
Warehouse	2	5	8	14
	3	4	3	10

It is required to find the units of commodity to be moved from the warehouses to the stores, such that the transportation costs are minimized. The maximum number of iterations allowed is 200.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     H03ABF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          MAMAX, MBMAX, M, MMM
      PARAMETER        (MAMAX=5,MBMAX=5,M=MAMAX+MBMAX,MMM=MAMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             Z
      INTEGER          I, IFAIL, J, L, MA, MB
*     .. Local Arrays ..
      INTEGER          K11(M), K12(M), K15(M), K6(M), K7(M), K8(M),
     +                 K9(M), KOST(MMM,MBMAX)
*     .. External Subroutines ..
      EXTERNAL         H03ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'H03ABF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) MA, MB
      IF (MA.GT.0 .AND. MA.LE.MAMAX .AND. MB.GT.0 .AND. MB.LE.MBMAX)
     +    THEN
         READ (NIN,*) (K15(I),I=1,MA+MB)
         DO 20 I = 1, MA
            READ (NIN,*) (KOST(I,J),J=1,MB)
   20    CONTINUE
         IFAIL = 0
*
         CALL H03ABF(KOST,MMM,MA,MB,MA+MB,K15,200,K7,K9,L,K6,K8,K11,K12,
     +               Z,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Total cost = ', Z
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Goods from  to'
         WRITE (NOUT,*)
         WRITE (NOUT,99998) (K11(I),K6(I),K8(I),I=1,MA+MB-1)
      END IF
      STOP
*
99999 FORMAT (1X,A,F5.1)
99998 FORMAT (1X,I3,I6,I5)
      END
```

9.2 Program Data

```
H03ABF Example Program Data
    3    3
    1    5    6    4    4    4
    8    8   11
    5    8   14
    4    3   10
```

9.3 Program Results

```
H03ABF Example Program Results

Total cost =  77.0

Goods from  to

  4     3    2
  2     3    3
  1     2    3
  1     1    3
  4     2    1
```