NAG Toolbox for MATLAB

h03ab

1 Purpose

h03ab solves the classical Transportation ('Hitchcock') problem.

2 Syntax

```
[k15, numit, k6, k8, k11, k12, z, ifail] = h03ab(kost, ma, k15, maxit, 'mb', mb, 'm', m)
```

3 Description

h03ab solves the Transportation problem by minimizing

$$z = \sum_{i}^{m_a} \sum_{j}^{m_b} c_{ij} x_{ij}.$$

subject to the constraints

$$\sum_{j}^{m_b} x_{ij} = A_i \qquad \text{(Availabilities)}$$

$$\sum_{i}^{m_a} \sum_{i} x_{ij} = B_j \qquad \text{(Requirements)}$$

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

h03ab uses the 'stepping stone' method, modified to accept degenerate cases.

4 References

Hadley G 1962 *Linear Programming* Addison–Wesley

5 Parameters

5.1 Compulsory Input Parameters

1: **kost**(**ldkost**,**mb**) – **int32 array**

ldkost, the first dimension of the array, must be at least **ma**.

The coefficients c_{ij}, for $i = 1, 2, \ldots, m_a$ and $j = 1, 2, \ldots, m_b$.

2: **ma** – **int32 scalar**

The number of sources, m_a.

Constraint: **ma** ≥ 1.

3: **k15**(**m**) – **int32 array**

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 $j = 1, 2, \ldots, m_b$.

4: **maxit – int32 scalar**

The maximum number of iterations allowed.

Constraint: **maxit** ≥ 1.

5.2 Optional Input Parameters

1: **mb – int32 scalar**

Default: The dimension of the array **kost**.

the number of destinations, m_b.

Constraint: **mb** ≥ 1.

2: **m – int32 scalar**

Default: The dimension of the arrays **k15**, **k6**, **k8**, **k11**, **k12**. (An error is raised if these dimensions are not equal.)

the value of $m_a + m_b$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldkost, k7, k9

5.4 Output Parameters

1: **k15**(**m**) **– int32 array**

The contents of **k15** are undefined.

2: **numit – int32 scalar**

The number of iterations performed.

3: **k6**(**m**) **– int32 array**

k6(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the source indices of the optimal solution (see **k11**).

4: **k8**(**m**) **– int32 array**

k8(k), for $k = 1, 2, \ldots, m_a + m_b - 1$, contains the destination indices of the optimal solution (see **k11**).

5: **k11**(**m**) **– int32 array**

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.

6: **k12**(**m**) **– int32 array**

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).

7: **z – double scalar**

The value of the minimized total cost.

8: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

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** $>$ **ldkost**.

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

```
kost = [int32(8), int32(8), int32(11);
     int32(5), int32(8), int32(14);
     int32(4), int32(3), int32(10)];
ma = int32(3);
k15 = [int32(1);
     int32(5);
     int32(6);
     int32(4);
     int32(4);
     int32(4)];
maxit = int32(200);
[k15Out, numit, k6, k8, k11, k12, z, ifail] = h03ab(kost, ma, k15, maxit)
```

```
k15Out =
           3
          10
          14
          11
           5
           0
numit =
           2
k6 =
           3
           3
           2
           1
           2
           3
k8 =
```

```
            2
            3
            3
            3
            1
            6
k11 =
            4
            2
            1
            1
            4
            2
k12 =
            3
           10
           14
           11
            5
            0
z =
    77
ifail =
            0
```