H02BFF – 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

H02BFF solves linear or integer programming problems specified in MPSX input format. It is not intended for large sparse problems.

2 Specification

```
 SUBROUTINE H02BFF(INFILE, MAXN, MAXM, OPTIM, XBLDEF, XBUDEF,
1                  MAXDPT, MSGLVL, N, M, X, CRNAME, IWORK, LIWORK,
2                  RWORK, LRWORK, IFAIL)
 INTEGER           INFILE, MAXN, MAXM, MAXDPT, MSGLVL, N, M,
1                  IWORK(LIWORK), LIWORK, LRWORK, IFAIL
 real              XBLDEF, XBUDEF, X(MAXN), RWORK(LRWORK)
 CHARACTER*3       OPTIM
 CHARACTER*8       CRNAME(MAXN+MAXM)
```

3 Description

H02BFF solves linear programming (LP) or integer programming (IP) problems specified in MPSX [1] input format. It calls either E04MFF (to solve an LP problem) or H02BBF and H02BZF (to solve an IP problem); these routines are designed to solve problems of the form

$$\underset{x\in R^n}{\text{minimize}}\; c^T x \;\text{ subject to }\; l \leq \begin{pmatrix} x \\ Ax \end{pmatrix} \leq u$$

where c is an n element vector and A is an m by n matrix (i.e., there are n variables and m general linear constraints). H02BBF is used if at least one of the variables is restricted to take an integer value at the optimum solution. The document for H02BUF should be consulted for a detailed description of the MPSX format.

In the MPSX data file the first free row, that is, a row defined with the row type N, is taken as the objective row. Similarly, if there are more than one RHS, RANGES or BOUNDS sets, then the first set is used for the optimization. H02BFF also prints the solution to the problem using the row and column names specified in the MPSX data file (by calling H02BVF).

4 References

[1] (1971) MPSX – Mathematical programming system *Program Number 5734 XM4* IBM Trade Corporation, New York

5 Parameters

1: INFILE — INTEGER *Input*

On entry: the unit number associated with the MPSX data file.

Constraint: $0 \leq$ INFILE ≤ 99.

2: MAXN — INTEGER *Input*

On entry: an upper limit for the number of variables in the problem.

Constraint: MAXN ≥ 1.

3: MAXM — INTEGER *Input*

On entry: an upper limit for the number of constraints (including the objective) in the problem.

Constraint: MAXM ≥ 1.

4: OPTIM — CHARACTER*3 *Input*

On entry: specifies the direction of the optimization. OPTIM must be set to 'MIN' for minimization and to 'MAX' for maximization.

Constraint: OPTIM = 'MIN' or 'MAX'.

5: XBLDEF — ***real*** *Input*

On entry: the default lower bound to be used for the variables in the problem, when none is specified in the BOUNDS section of the MPSX data file. For a standard LP or IP problem XBLDEF would normally be set to zero.

6: XBUDEF — ***real*** *Input*

On entry: the default upper bound to be used for the variables in the problem, when none is specified in the BOUNDS section of the MPSX data file. For a standard LP or IP problem XBUDEF would normally be set to 'infinity' (i.e., XBUDEF $\geq 10^{20}$).

Constraint: XBUDEF $\geq$ XBLDEF.

7: MAXDPT — INTEGER *Input*

On entry: for an IP problem, MAXDPT must specify the maximum depth of the branch and bound tree.

Constraint: MAXDPT ≥ 2.

For an LP problem, MAXDPT is not referenced.

8: MSGLVL — INTEGER *Input*

On entry: the amount of printout produced by E04MFF or H02BBF, as indicated below. For a description of the printed output see Section 8.2 of the document for E04MFF or Section 5.1 of the document for H02BBF (as appropriate). All output is written to the current advisory message unit (as defined by X04ABF).

For an LP problem (E04MFF):

Value	**Definition**
0	No output.
1	The final solution only.
5	One line of output for each iteration (no printout of the final solution).
10	The final solution and one line of output for each iteration.

For an IP problem (H02BBF):

Value	**Definition**
0	No output.
1	The final IP solution only.
5	One line of output for each node investigated and the final IP solution.
10	The original LP solution (first node) with dummy names for the rows and columns, one line of output for each node investigated and the final IP solution with MPSX names for the rows and columns.

9: N — INTEGER *Output*

On exit: n, the actual number of variables in the problem.

10: M — INTEGER *Output*

On exit: m, the actual number of general linear constraints in the problem.

11: X(MAXN) — ***real*** array *Output*

On exit: the solution to the problem, stored in X(1),X(2),...,X(N). X(i) is the value of the variable whose MPSX name is stored in CRNAME(i), for $i = 1, 2, \ldots$,N.

12: CRNAME(MAXN+MAXM) — CHARACTER*8 array *Output*

On exit: the first N elements contain the MPSX names for the variables in the problem.

13: IWORK(LIWORK) — INTEGER array *Output*

On exit: the first (N+M) elements contain ISTATE (the status of the constraints in the working set at the solution). Further details can be found in Section 5 of the document for E04MFF or Section 5 of the document for H02BZF (as appropriate).

14: LIWORK — INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which H02BFF is called.

Constraints:

For an LP problem, LIWORK $\geq$ 4 $\times$ MAXN + MAXM + 3.
For an IP problem, LIWORK $\geq$ (25+MAXN+MAXM) $\times$ MAXDPT + 7 $\times$ MAXN + 2 $\times$ MAXM + 4.

15: RWORK(LRWORK) — ***real*** array *Output*

On exit: the first (N+M) elements contain BL (the lower bounds), the next (N+M) elements contain BU (the upper bounds) and the next (N+M) elements contain CLAMDA (the Lagrange multipliers). Further details can be found in Section 5 of the document for E04MFF or Section 5 of the document for H02BZF (as appropriate). Note that for an IP problem the contents of BL and BU may not be the same as those originally specified by the user in the MPSX data file and/or via the parameters XBLDEF and XBUDEF.

16: LRWORK — INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which H02BFF is called.

Constraints:

For an LP problem, LRWORK $\geq$ 2 $\times$ MIN(MAXN,MAXM+1)^2 + MAXM $\times$ MAXN + 12 $\times$ MAXN + 9 $\times$ MAXM.
For an IP problem, LRWORK $\geq$ MAXDPT $\times$ (MAXN+1) + 2 $\times$ MIN(MAXN,MAXM+1)^2 + MAXM $\times$ MAXN + 19 $\times$ MAXN + 15 $\times$ MAXM.

17: 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

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL $= i < 0$

Either MAXM and/or MAXN are too small or the MPSX data file is non-standard and/or corrupt. This corresponds to IFAIL $= -i$ in Section 6 of the document for H02BUF.

IFAIL = 1

X is a weak local minimum. This means that the solution is not unique.

IFAIL = 2

The solution appears to be unbounded. This value of IFAIL implies that a step as large as XBUDEF would have to be taken in order to continue the algorithm. See Section 8.

IFAIL = 3

No feasible point was found, i.e., it was not possible to satisfy all the constraints to within the feasibility tolerance (defined internally as $\sqrt{\text{machine precision}}$). See Section 8.

IFAIL = 4

The maximum number of iterations (defined internally as $\max(50{,}5(n+m))$) was reached before normal termination occurred. See Section 8.

IFAIL = 5

An input parameter is invalid. Refer to the printed output to determine which parameter must be re-defined.

IFAIL = 6

A serious error has occurred in an internal call to either E04MFF or H02BBF (as appropriate). Check all subroutine calls and array dimensions.

For an IP problem only:

IFAIL = 7

The solution reported is not the optimum solution. See Section 8.

IFAIL = 8

MAXDPT is too small. Try increasing its value (along with that of LIWORK and/or LRWORK if appropriate) and rerun H02BFF.

IFAIL = 9

No feasible integer point was found, i.e., it was not possible to satisfy all the integer variables to within the integer feasibility tolerance (defined internally as 10^{-5}). See Section 8.

7 Accuracy

The routine implements a numerically stable active set strategy and returns solutions that are as accurate as the condition of the problem warrants on the machine.

8 Further Comments

For an LP problem only:

if IFAIL = 2 on exit, users can obtain more information by making separate calls to H02BUF, E04MFF and H02BVF (in that order). Note that this will (by default) cause the final LP solution to be printed twice on the current advisory message unit (see X04ABF), once with dummy names for the rows and columns and once with user supplied names. To suppress the printout of the final LP solution with dummy names for the rows and columns, include the statement

```
CALL E04MHF(' Print Level = 5 ')
```

prior to calling E04MFF.

if IFAIL = 3 on exit, users are recommended to reset the value of the feasibility tolerance and rerun H02BFF. (Further advice is given under the description of IFAIL = 3 in Section 6 of the document for E04MFF). For example, to reset the value of the feasibility tolerance to 0.01, include the statement

```
CALL E04MHF(' Feasibility Tolerance = 0.01 ')
```

prior to calling H02BFF.

if IFAIL = 4 on exit, users are recommended to increase the maximum number of iterations allowed before termination and rerun H02BFF. For example, to increase the maximum number of iterations to 500, include the statement

```
CALL E04MHF(' Iteration Limit = 500 ')
```

prior to calling H02BFF.

Note that H02BUF uses an 'infinite' bound size of 10^{20} in the definition of l and u. In other words, any element of u greater than or equal to 10^{20} will be regarded as $+\infty$ (and similarly any element of l less than or equal to -10^{20} will be regarded as $-\infty$). If this value is deemed to be 'inappropriate', users are recommended to reset the value of the 'infinite' bound size and make any necessary changes to BL and/or BU prior to calling E04MFF. For example, to reset the value of the 'infinite' bound size to 10000, include the statement

```
CALL E04MHF(' Infinite Bound Size = 1.0E+4 ')
```

prior to calling E04MFF.

For an IP problem only:

if IFAIL = 2,3,4,7 or 9 on exit, users can obtain more information by making separate calls to H02BUF, H02BBF, H02BZF and H02BVF (in that order).

Note that H02BUF uses an 'infinite' bound size of 10^{20} in the definition of l and u. In other words, any element of u greater than or equal to 10^{20} will be regarded as $+\infty$ (and similarly any element of l less than or equal to -10^{20} will be regarded as $-\infty$). If this value is deemed to be 'inappropriate', users are recommended to reset the value of the parameter BIGBND (as described in Section 5 of the document for H02BBF) and make any necessary changes to BL and/or BU prior to calling H02BBF.

9 Example

This example solves the same problem as the example for H02BUF, except that it treats it as an IP problem.

One of the applications of integer programming is to the so-called diet problem. Given the nutritional content of a selection of foods, the cost of each food, the amount available of each food and the consumer's minimum daily energy requirements, the problem is to find the cheapest combination. This gives rise to the following problem:

minimize

$$c^T x$$

subject to

$$Ax \geq b,$$
$$0 \leq x \leq u,$$

where

$c = (3\ 24\ 13\ 9\ 20\ 19)^T,\ x = (x_1, x_2, x_3, x_4, x_5, x_6)^T,$

x_1, x_2 and x_6 are real,

x_3, x_4 and x_5 are integer,

$$A = \begin{pmatrix} 110 & 205 & 160 & 160 & 420 & 260 \\ 4 & 32 & 13 & 8 & 4 & 14 \\ 2 & 12 & 54 & 285 & 22 & 80 \end{pmatrix}, \quad b = \begin{pmatrix} 2000 \\ 55 \\ 800 \end{pmatrix} \text{ and}$$

$u = (4\ 3\ 2\ 8\ 2\ 2)^T.$

The rows of A correspond to energy, protein and calcium and the columns of A correspond to oatmeal, chicken, eggs, milk, pie and bacon respectively.

The MPSX data representation of this problem is given in Section 9.2.

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.

```
*     H02BFF Example Program Text
*     Mark 18 Revised.  NAG Copyright 1997.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MAXN, MAXM
      PARAMETER        (MAXN=50,MAXM=50)
      real             XBLDEF, XBUDEF
      PARAMETER        (XBLDEF=0.0e0,XBUDEF=1.0e+20)
      INTEGER          MAXDPT
      PARAMETER        (MAXDPT=3*MAXN/2)
      INTEGER          MSGLVL
      PARAMETER        (MSGLVL=5)
      INTEGER          LIWORK
      PARAMETER        (LIWORK=(25+MAXN+MAXM)*MAXDPT+2*MAXM+7*MAXN+4)
      INTEGER          LRWORK
      PARAMETER        (LRWORK=MAXDPT*(MAXN+1)
     +                 +2*MAXN**2+MAXM*MAXN+19*MAXN+15*MAXM)
      CHARACTER*3      OPTIM
      PARAMETER        (OPTIM='MIN')
*     .. Local Scalars ..
      INTEGER          IFAIL, INFILE, M, N
*     .. Local Arrays ..
      real             RWORK(LRWORK), X(MAXN)
      INTEGER          IWORK(LIWORK)
      CHARACTER*8      CRNAME(MAXN+MAXM)
*     .. External Subroutines ..
      EXTERNAL         H02BFF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'H02BFF Example Program Results'
```

```
*     Skip heading in data file
      READ (NIN,*)
*
*     Solve the problem
*
      INFILE = NIN
      IFAIL = 0
*
      CALL H02BFF(INFILE,MAXN,MAXM,OPTIM,XBLDEF,XBUDEF,MAXDPT,MSGLVL,N,
     +            M,X,CRNAME,IWORK,LIWORK,RWORK,LRWORK,IFAIL)
*
      STOP
      END
```

9.2 Program Data

```
H02BFF Example Program Data
NAME          DIET
ROWS
 G  ENERGY
 G  PROTEIN
 G  CALCIUM
 N  COST
COLUMNS
    OATMEAL   ENERGY      110.0
    OATMEAL   PROTEIN     4.0
    OATMEAL   CALCIUM     2.0
    OATMEAL   COST        3.0
    CHICKEN   ENERGY      205.0
    CHICKEN   PROTEIN     32.0
    CHICKEN   CALCIUM     12.0
    CHICKEN   COST        24.0
    INTEGER   'MARKER'                 'INTORG'
    EGGS      ENERGY      160.0
    EGGS      PROTEIN     13.0
    EGGS      CALCIUM     54.0
    EGGS      COST        13.0
    MILK      ENERGY      160.0
    MILK      PROTEIN     8.0
    MILK      CALCIUM     285.0
    MILK      COST        9.0
    PIE       ENERGY      420.0
    PIE       PROTEIN     4.0
    PIE       CALCIUM     22.0
    PIE       COST        20.0
    INTEGER   'MARKER'                 'INTEND'
    BACON     ENERGY      260.0
    BACON     PROTEIN     14.0
    BACON     CALCIUM     80.0
    BACON     COST        19.0
RHS
    DEMANDS   ENERGY      2000.0
    DEMANDS   PROTEIN     55.0
    DEMANDS   CALCIUM     800.0
```

```
BOUNDS
 UI SERVINGS  OATMEAL     4.0
 UI SERVINGS  CHICKEN     3.0
 UP SERVINGS  EGGS        2.0
 UP SERVINGS  MILK        8.0
 UP SERVINGS  PIE         2.0
 UI SERVINGS  BACON       2.0
ENDATA
```

9.3 Program Results

```
H02BFF Example Program Results

*** H02BBF
*** Start of NAG Library implementation details ***

Implementation title: Generalised Base Version
           Precision: FORTRAN double precision
        Product Code: FLBAS18D
                Mark: 18A

*** End of NAG Library implementation details ***

Parameters
----------

Linear constraints......        3       First integer solution..      OFF
Variables...............        6       Max depth of the tree...       75

Feasibility tolerance...  1.05E-08       Print level............        5
Infinite bound size.....  1.00E+20       EPS (machine precision).  1.11E-16

Integer feasibility tol.  1.00E-05       Iteration limit.........       50
Max number of nodes.....      NONE

** Workspace provided with MAXDPT =    75: LRWORK = 10075  LIWORK =  9679
** Workspace required with MAXDPT =    75: LRWORK =   677  LIWORK =  2587


  *** Optimum LP solution ***    92.50000


  *** Start of tree search ***


 Node  Parent    Obj       Varbl  Value      Lower      Upper      Value    Depth
  No    Node    Value      Chosen Before     Bound      Bound      After
   2      1      93.2         4    4.50      5.00       8.00       5.00       1
   3      1      93.8         4    4.50     0.000E+00   4.00       4.00       1
   4      2      94.8         5    1.81      2.00       2.00       2.00       2
   5      2      96.1         5    1.81     0.000E+00   1.00       1.00       2
   6      3      96.9         6   0.308      1.00       2.00       1.00       2
   7      3      94.5         6   0.308     0.000E+00  0.000E+00  0.000E+00   2
   8      7      96.5         3   0.500      1.00       2.00       1.00       3
   9      7      97.4         3   0.500     0.000E+00  0.000E+00  0.000E+00   3
  10      4      97.0         1    3.27      4.00       4.00       4.00       3
    *** Integer solution ***
```

```
 Node  Parent    Obj       Varbl  Value      Lower     Upper     Value    Depth
  No    Node    Value      Chosen Before     Bound     Bound     After
  11     4      95.7          1    3.27     0.000E+00   3.00      3.00       3
  12    11      99.5     CO   4    5.19      6.00       8.00      6.00       4
  13    11      96.2          4    5.19      5.00       5.00      5.00       4
  14     5      97.3     CO   4    7.13      8.00       8.00      8.00       3
  15     5      96.5          4    7.13      5.00       7.00      7.00       3
  16    13      107.     CO   6   0.115      1.00       2.00      1.00       5
  17    13      96.4          6   0.115     0.000E+00  0.000E+00 0.000E+00   5
  18    17      103.     CO   3   0.188      1.00       2.00      1.00       6
  19    17      97.5     CO   3   0.188     0.000E+00  0.000E+00 0.000E+00   6
  20    15      101.     CO   6   0.769E-01  1.00       2.00      1.00       4
  21    15      96.6          6   0.769E-01 0.000E+00  0.000E+00 0.000E+00   4
  22     8      97.2     CO   4    3.50      4.00       4.00      4.00       4
  23     8      98.5     CO   4    3.50     0.000E+00   3.00      3.00       4
  24    21      100.     CO   3   0.125      1.00       2.00      1.00       5
  25    21      97.3     CO   3   0.125     0.000E+00  0.000E+00 0.000E+00   5
  26     6      97.0     CO   4    2.88      3.00       4.00      3.00       3
  27     6      105.     CO   4    2.88     0.000E+00   2.00      2.00       3
   *** End of tree search ***


Total of    27 nodes investigated.

Exit H02BBF - Optimum IP solution found.

Final IP objective value =    97.00000


Varbl   State     Value      Lower Bound   Upper Bound    Lagr Mult   Residual

OATMEAL  EQ    4.00000       4.00000       4.00000        3.000      0.0000E+00
CHICKEN  LL   0.000000E+00  0.000000E+00   3.00000        24.00      0.0000E+00
EGGS     LL   0.000000E+00  0.000000E+00   2.00000        13.00      0.0000E+00
MILK     LL    5.00000       5.00000       8.00000        9.000      0.0000E+00
PIE      EQ    2.00000       2.00000       2.00000        20.00      0.0000E+00
BACON    LL   0.000000E+00  0.000000E+00   2.00000        19.00      0.0000E+00


L Con   State     Value      Lower Bound   Upper Bound    Lagr Mult   Residual

ENERGY   FR    2080.00       2000.00          None      0.0000E+00   80.00
PROTEIN  FR    64.0000       55.0000          None      0.0000E+00   9.000
CALCIUM  FR    1477.00       800.000          None      0.0000E+00   677.0
```