NAG Toolbox for MATLAB

h02bb

1 Purpose

h02bb solves 'zero-one', 'general', 'mixed' or 'all' integer programming problems using a branch and bound method. The function may also be used to find either the first integer solution or the optimum integer solution. It is not intended for large sparse problems.

2 Syntax

```
[itmax, toliv, tolfes, bigbnd, x, objmip, iwork, rwork, ifail] =
h02bb(itmax, msglvl, a, bl, bu, intvar, cvec, maxnod, intfst, toliv,
tolfes, bigbnd, x, 'n', n, 'm', m, 'maxdpt', maxdpt)
```

3 Description

h02bb is capable of solving certain types of integer programming (IP) problems using a branch and bound (B&B) method, see Taha 1987. In order to describe these types of integer programs and to briefly state the B&B method, we define the following linear programming (LP) problem:

Minimize

$$F(x) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

subject to

$$\sum_{j=1}^{n} a_{ij}x_j \left\{ \begin{array}{c} = \\ \leq \\ \geq \end{array} \right\} b_i, \qquad i = 1, 2, \ldots, m$$

$$l_j \leq x_j \leq u_j, \qquad j = 1, 2, \ldots, n \tag{1}$$

If, in (1), it is required that (some or) all the variables take integer values, then the integer program is of type (*mixed* or) *all* general IP problem. If additionally, the integer variables are restricted to take only 0–1 values (i.e., $l_j = 0$ and $u_j = 1$) then the integer program is of type (mixed or all) *zero-one* IP problem.

The B&B method applies directly to these integer programs. The general idea of B&B (for a full description see Dakin 1965 or Mitra 1973) is to solve the problem without the integral restrictions as an LP problem (first *node*). If in the optimal solution an integer variable x_k takes a noninteger value x_k^*, two LP sub-problems are created by *branching*, imposing $x_k \leq [x_k^*]$ and $x_k \geq [x_k^*] + 1$ respectively, where $[x_k^*]$ denotes the integer part of x_k^*. This method of branching continues until the first integer solution (*bound*) is obtained. The hanging nodes are then solved and investigated in order to prove the optimality of the solution. At each node, an LP problem is solved using e04mf.

4 References

Dakin R J 1965 A tree search algorithm for mixed integer programming problems *Comput. J.* **8** 250–255

Mitra G 1973 Investigation of some branch and bound strategies for the solution of mixed integer linear programs *Math. Programming* **4** 155–170

Taha H A 1987 *Operations Research: An Introduction* Macmillan, New York

5 Parameters

5.1 Compulsory Input Parameters

1: **itmax – int32 scalar**

An upper bound on the number of iterations for each LP problem.

2: **msglvl – int32 scalar**

The amount of printout produced by h02bb.

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), one line of output for each node investigated and the final IP solution.

3: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{m})$

The second dimension of the array must be at least **n** if $\mathbf{m} > 0$ and at least 1 if $\mathbf{m} = 0$

The ith row of **a** must contain the coefficients of the ith general constraint, for $i = 1, 2, \ldots, m$.

If $\mathbf{m} = 0$ then the array **a** is not referenced.

4: **bl($\mathbf{n} + \mathbf{m}$) – double array**
5: **bu($\mathbf{n} + \mathbf{m}$) – double array**

bl must contain the lower bounds and **bu** the upper bounds, for all the constraints in the following order. The first n elements of each array must contain the bounds on the variables, and the next m elements the bounds for the general linear constraints (if any). To specify a nonexistent lower bound (i.e., $l_j = -\infty$), set $\mathbf{bl}(j) \leq -\mathbf{bigbnd}$ and to specify a nonexistent upper bound (i.e., $u_j = +\infty$), set $\mathbf{bu}(j) \geq \mathbf{bigbnd}$. To specify the jth constraint as an equality, set $\mathbf{bl}(j) = \mathbf{bu}(j) = \beta$, say, where $|\beta| < \mathbf{bigbnd}$.

Constraints:

$\mathbf{bl}(j) \leq \mathbf{bu}(j)$, for $j = 1, 2, \ldots, \mathbf{n} + \mathbf{m}$;
if $\mathbf{bl}(j) = \mathbf{bu}(j) = \beta$, $|\beta| < \mathbf{bigbnd}$.

6: **intvar(n) – int32 array**

Indicates which are the integer variables in the problem. For example, if x_j is an integer variable then **intvar**(j) must be set to 1, and 0 otherwise.

Constraints:

$\mathbf{intvar}(j) = 0$ or 1, for $j = 1, 2, \ldots, \mathbf{n}$;
$\mathbf{intvar}(j) = 1$ for at least one value of j.

7: **cvec(n) – double array**

The coefficients c_j of the objective function $F(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$. The function attempts to find a minimum of $F(x)$. If a maximum of $F(x)$ is desired, **cvec**(j) should be set to $-c_j$, for $j = 1, 2, \ldots, n$, so that the function will find a minimum of $-F(x)$.

8: **maxnod – int32 scalar**

The maximum number of nodes that are to be searched in order to find a solution (optimum integer solution). If $\mathbf{maxnod} \leq 0$ and $\mathbf{intfst} \leq 0$, then the B&B tree search is continued until all the nodes have been investigated.

9: **intfst – int32 scalar**

Specifies whether to terminate the B&B tree search after the first integer solution (if any) is obtained. If $\mathbf{intfst} > 0$ then the B&B tree search is terminated at node k say, which contains the first integer solution. For $\mathbf{maxnod} > 0$ this applies only if $k \leq \mathbf{maxnod}$.

10: **toliv – double scalar**

The integer feasibility tolerance; i.e., an integer variable is considered to take an integer value if its violation does not exceed **toliv**. For example, if the integer variable x_j is near unity then x_j is considered to be integer only if $(1 - \mathbf{toliv}) \leq x_j \leq (1 + \mathbf{toliv})$.

11: **tolfes – double scalar**

The maximum acceptable absolute violation in each constraint at a ‘feasible’ point (feasibility tolerance); i.e., a constraint is considered satisfied if its violation does not exceed **tolfes**.

12: **bigbnd – double scalar**

The ‘infinite’ bound size in the definition of the problem constraints. More precisely, any upper bound greater than or equal to **bigbnd** will be regarded as $+\infty$ and any lower bound less than or equal to $-$**bigbnd** will be regarded as $-\infty$.

13: **x(n) – double array**

An initial estimate of the original LP solution.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **bl**, **bu**, **cvec**, **x**. (An error is raised if these dimensions are not equal.)

n, the number of variables.

Constraint: $\mathbf{n} > 0$.

2: **m – int32 scalar**

m, the number of general linear constraints.

Constraint: $\mathbf{m} \geq 0$.

3: **maxdpt – int32 scalar**

The maximum depth of the B&B tree used for branch and bound.

Suggested value: $\mathbf{maxdpt} = 3 \times \mathbf{n}/2$.

Default: $3 \times \mathbf{n}/2$

Constraint: $\mathbf{maxdpt} \geq 2$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, liwork, lrwork

5.4 Output Parameters

1: **itmax – int32 scalar**

Unchanged if on entry $\mathbf{itmax} > 0$, else $\mathbf{itmax} = \max(50, 5 \times (\mathbf{n} + \mathbf{m}))$.

2: **toliv – double scalar**

Unchanged if on entry $\mathbf{toliv} > 0.0$, else $\mathbf{toliv} = 10^{-5}$.

3: **tolfes – double scalar**

Unchanged if on entry $\mathbf{tolfes} > 0.0$, else $\mathbf{tolfes} = \sqrt{\epsilon}$ (where ϵ is the ***machine precision***).

4: **bigbnd – double scalar**

Unchanged if on entry **bigbnd** > 0.0, else **bigbnd** $= 10^{20}$.

5: **x**(**n**) **– double array**

With **ifail** $= 0$, **x** contains a solution which will be an estimate of either the optimum integer solution or the first integer solution, depending on the value of **intfst**. If **ifail** $= 9$, then **x** contains a solution which will be an estimate of the best integer solution that was obtained by searching **maxnod** nodes.

6: **objmip – double scalar**

With **ifail** $=$ 0 or 9, **objmip** contains the value of the objective function for the IP solution.

7: **iwork**(**liwork**) **– int32 array**

8: **rwork**(**lrwork**) **– double array**

9: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Note: h02bb may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

No feasible integer point was found, i.e., it was not possible to satisfy all the integer variables to within the integer feasibility tolerance (determined by **toliv**). Increase **toliv** and rerun h02bb.

ifail $= 2$

The original LP solution appears to be unbounded. This value of **ifail** implies that a step as large as **bigbnd** would have to be taken in order to continue the algorithm (see Section 8).

ifail $= 3$

No feasible point was found for the original LP problem, i.e., it was not possible to satisfy all the constraints to within the feasibility tolerance (determined by **tolfes**). If the data for the constraints are accurate only to the absolute precision σ, you should ensure that the value of the feasibility tolerance is greater than σ. For example, if all elements of A are of order unity and are accurate only to three decimal places, the feasibility tolerance should be at least 10^{-3} (see Section 8).

ifail $= 4$

The maximum number of iterations (determined by **itmax**) was reached before normal termination occurred for the original LP problem (see Section 8).

ifail $= 5$

Not used by this function.

ifail $= 6$

An input parameter is invalid.

ifail $= 7$

The IP solution reported is not the optimum IP solution. In other words, the B&B tree search for at least one of the branches had to be terminated since an LP sub-problem in the branch did not have a solution (see Section 8).

ifail = 8

The maximum depth of the B&B tree used for branch and bound (determined by **maxdpt**) is too small. Increase **maxdpt** (along with **liwork** and/or **lrwork** if appropriate) and rerun h02bb.

ifail = 9

The IP solution reported is the best IP solution for the number of nodes (determined by **maxnod**) investigated in the B&B tree.

ifail = 10

No feasible integer point was found for the number of nodes (determined by **maxnod**) investigated in the B&B tree.

overflow

It may be possible to avoid the difficulty by increasing the magnitude of the feasibility tolerance (**tolfes**) and rerunning the program. If the message recurs even after this change, see Section 8.

7 Accuracy

h02bb 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

The original LP problem may not have an optimum solution, i.e., h02bb terminates with **ifail** = 2, 3 or 4 or overflow may occur. In this case, you are recommended to relax the integer restrictions of the problem and try to find the optimum LP solution by using e04mf instead.

In the B&B method, it is possible for an LP sub-problem to terminate without finding a solution. This may occur due to the number of iterations exceeding the maximum allowed. Therefore the B&B tree search for that particular branch cannot be continued. Thus the final IP solution reported is not the optimum IP solution (see **ifail** = 7). For the second and unlikely case, a solution could not be found despite a second attempt at an LP solution.

9 Example

```
itmax = int32(0);
msglvl = int32(10);
a = [2, 5;
     2, -2;
     3, 2];
bl = [0;
     0;
     -1e+20;
     -1e+20;
     5];
bu = [1e+20;
     1e+20;
     15;
     5;
     1e+20];
intvar = [int32(1);
     int32(1)];
cvec = [-3;
     -4];
maxnod = int32(0);
intfst = int32(0);
toliv = 0;
tolfes = 0;
```

```
bigbnd = 1e+20;
x = [1;
     1];
[itmaxOut, tolivOut, tolfesOut, bigbndOut, xOut, objmip, iwork, rwork,
ifail] = ...
     h02bb(itmax, msglvl, a, bl, bu, intvar, cvec, maxnod, intfst, toliv,
...
     tolfes, bigbnd, x, 'maxdpt', int32(4));
```

```
*** H02BBF

 Parameters
 ----------

 Linear constraints......            3          First integer solution..
OFF
 Variables..............            2          Max depth of the tree...
4

 Feasibility tolerance...   1.05E-08           Print level............
10
 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 =      4: LRWORK =     84   LIWORK =
137
 ** Workspace required with MAXDPT =      4: LRWORK =     84   LIWORK =
137


   *** Optimum LP solution ***   -17.50000


 Varbl State       Value       Lower Bound    Upper Bound     Lagr Mult
Residual

 V   1     FR     3.92857         0.00000           None           0.000
3.929
 V   2     FR     1.42857         0.00000           None           0.000
1.429


 L Con State       Value       Lower Bound    Upper Bound     Lagr Mult
Residual

 L   1     UL     15.0000            None        15.0000        -1.000
0.000
 L  2    UL    5.00000          None         5.00000      -0.5000      -
8.8818E-16
 L   3     FR     14.6429         5.00000           None           0.000
9.643


   *** Start of tree search ***


  Node  Parent    Obj        Varbl  Value       Lower      Upper      Value
Depth
   No    Node    Value      Chosen Before      Bound      Bound      After
    2     1    No Feas Soln     1     3.93       4.00        None       4.00
1
    3     1     -16.2           1     3.93       0.00        3.00       3.00
1
    4     3     -15.5           2     1.80       2.00        None       2.00
2
    5     3     -13.0           2     1.80       0.00        1.00       1.00
```

```
2
      *** Integer solution ***


  Node  Parent    Obj       Varbl  Value      Lower     Upper     Value
Depth
   No    Node    Value      Chosen Before     Bound     Bound     After
    6      4    No Feas Soln     1     2.50      3.00      3.00      3.00
3
    7      4     -14.8          1     2.50      0.00      2.00      2.00
3
    8      7     -12.0     CO    2     2.20      3.00      None      3.00
4
    9      7     -14.0          2     2.20      2.00      2.00      2.00
4
      *** Integer solution ***

     *** End of tree search ***


 Total of     9 nodes investigated.

 Exit H02BBF - Optimum IP solution found.

 Final IP objective value =   -14.00000



 Varbl  State       Value       Lower  Bound    Upper  Bound      Lagr  Mult
Residual

 V   1      UL      2.00000          0.00000          2.00000         -3.000
0.000
 V   2      EQ      2.00000          2.00000          2.00000         -4.000
0.000


 L  Con  State       Value       Lower  Bound    Upper  Bound      Lagr  Mult
Residual

 L   1      FR      14.0000            None          15.0000          0.000
1.000
 L   2      FR      0.00000            None          5.00000          0.000
5.000
 L   3      FR      10.0000          5.00000            None          0.000
5.000
```