

NAG Toolbox

nag_mip_ilp_info (h02bz)

1 Purpose

nag_mip_ilp_info (h02bz) extracts more information associated with the solution of an integer programming problem computed by nag_mip_ilp_dense (h02bb).

2 Syntax

```
[bl, bu, clamda, istrate, ifail] = nag_mip_ilp_info(n, m, iwork, rwork)
[bl, bu, clamda, istrate, ifail] = h02bz(n, m, iwork, rwork)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: *liwork* and *lrwork* were removed from the interface.

3 Description

nag_mip_ilp_info (h02bz) extracts the following information associated with the solution of an integer programming problem computed by nag_mip_ilp_dense (h02bb). The upper and lower bounds used for the solution, the Lagrange-multipliers (costs), and the status of the variables at the solution.

In the branch and bound method employed by nag_mip_ilp_dense (h02bb), the arrays **bl** and **bu** are used to impose restrictions on the values of the integer variables in each sub-problem. That is, if the variable x_j is restricted to take value v_j in a particular sub-problem, then $\mathbf{bl}(j) = \mathbf{bu}(j) = v_j$ is set in the sub-problem. Thus, on exit from this function, some of the elements of **bl** and **bu** which correspond to integer variables may contain these imposed values, rather than those originally supplied to nag_mip_ilp_dense (h02bb).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **n** – INTEGER

This **must** be the same argument **n** as supplied to nag_mip_ilp_dense (h02bb).

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

2: **m** – INTEGER

This **must** be the same argument **m** as supplied to nag_mip_ilp_dense (h02bb).

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

3: **iwork**(*liwork*) – INTEGER array

This **must** be the same argument **iwork** as supplied to nag_mip_ilp_dense (h02bb). It is used to pass information from nag_mip_ilp_dense (h02bb) to nag_mip_ilp_info (h02bz) and therefore the contents of this array **must not** be changed before calling nag_mip_ilp_info (h02bz).

4: **rwork**(*lrwork*) – REAL (KIND=nag_wp) array

This **must** be the same argument **rwork** as supplied to nag_mip_ilp_dense (h02bb). It is used to pass information from nag_mip_ilp_dense (h02bb) to nag_mip_ilp_info (h02bz) and therefore the contents of this array **must not** be changed before calling nag_mip_ilp_info (h02bz).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **bl**(**n + m**) – REAL (KIND=nag_wp) array

If nag_mip_ilp_dense (h02bb) exits with **ifail** = 0, 7 or 9, the values in the array **bl** contain the lower bounds imposed on the integer solution for all the constraints. The first **n** elements contain the lower bounds on the variables, and the next **m** elements contain the lower bounds for the general linear constraints (if any).

2: **bu**(**n + m**) – REAL (KIND=nag_wp) array

If nag_mip_ilp_dense (h02bb) exits with **ifail** = 0, 7 or 9, the values in the array **bu** contain the upper bounds imposed on the integer solution for all the constraints. The first **n** elements contain the upper bounds on the variables, and the next **m** elements contain the upper bounds for the general linear constraints (if any).

3: **clamda**(**n + m**) – REAL (KIND=nag_wp) array

If nag_mip_ilp_dense (h02bb) exits with **ifail** = 0, 7 or 9, the values in the array **clamda** contain the values of the Lagrange-multipliers for each constraint with respect to the current working set. The first **n** elements contain the multipliers (reduced costs) for the bound constraints on the variables, and the next **m** elements contain the multipliers (shadow costs) for the general linear constraints (if any).

4: **istate**(**n + m**) – INTEGER array

If nag_mip_ilp_dense (h02bb) exits with **ifail** = 0, 7 or 9, the values in the array **istate** indicate the status of the constraints in the working set at an integer solution. Otherwise, **istate** indicates the composition of the working set at the final iterate. The significance of each possible value of **istate**(*j*) is as follows.

istate (<i>j</i>)	Meaning
-2	The constraint violates its lower bound by more than tolfes (the feasibility tolerance, see h02bb).
-1	The constraint violates its upper bound by more than tolfes .
0	The constraint is satisfied to within tolfes , but is not in the working set.
1	This inequality constraint is included in the working set at its lower bound.
2	This inequality constraint is included in the working set at its upper bound.
3	This constraint is included in the working set as an equality. This value of istate can occur only when bl (<i>j</i>) = bu (<i>j</i>).
4	This corresponds to an integer solution being declared with x_j being temporarily fixed at its current value. This value of istate can occur only when ifail = 0, 7 or 9 on exit from h02bb.

5: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, $\mathbf{n} \leq 0$,
or $\mathbf{m} < 0$.

ifail = -99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = -399

Your licence key may have expired or may not have been installed correctly.

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

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 nutritional 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

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

is integer,

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

and

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

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 following program solves the above problem to obtain the optimal integer solution and then examines the effect of increasing the energy required to 2200 units.

9.1 Program Text

```

function h02bz_example

fprintf('h02bz example results\n\n');

% Find vector x of length n that Minimizes c.x subject to:
% bl(1:n) <= x <= bu(1:n) and m linear constraints
% bl(n+1:n+m) <= Ax <= bu(n+1:n_m)
n = nag_int(6);
m = nag_int(3);
a = [110, 205, 160, 160, 420, 260;
     4, 32, 13, 8, 4, 14;
     2, 12, 54, 285, 22, 80];
bl = [ 0; 0; 0; 0; 0; 0; 2000; 55; 800];
bu = [ 4; 3; 2; 8; 2; 2; 1e+20; 1e+20; 1e+20];

% All of x is integer.
intvar = nag_int([1; 1; 1; 1; 1; 1]);
c = [3; 24; 13; 9; 20; 19];
% Initial guess
x = zeros(6,1);

itmax = nag_int(0);
msglvl = nag_int(0);
maxnod = nag_int(0);
intfst = nag_int(0);
toliv = 0;
tolfes = 0;
bigbnd = 1e+20;
[itmax, toliv, tolfes, bigbnd, x, objmip, iwork, rwork, ifail] = ...
h02bb( ...
    itmax, msglvl, a, bl, bu, intvar, c, maxnod, ...
    intfst, toliv, tolfes, bigbnd, x);

% Extract further solution information and display
[bl, bu, clamda, istate, ifail] = ...
h02bz(n, m, iwork, rwork);

fprintf('Final IP objective value = %11.4f\n\n',objmip)

fprintf('%6s%8s%8s%16s%14s%13s\n','Varbl','State','Value',...
        'Lower Bound','Upper Bound','Lagr Mult');

chstate = ['VL';'VU';'FR';'LL';'UU';'EQ';'TF'];
vch = ['Oatmeal';'Chicken';'Eggs';'Milk';'Pie';'Bacon'];
cch = ['Energy';'Protein';'Calcium'];

for i=1:n
    ich = double(istate(i)) + 3;
    fprintf('%7s%6s%10.2f%14.2f%14.2f%13.2f\n',vch(i,:),...
            chstate(ich,:),x(i),bl(i),bu(i),clamda(i))
end

fprintf('\n%6s%8s%8s%16s%14s%13s\n','L Con','State','Value',...
        'Lower Bound','Upper Bound','Lagr Mult');
y = a*x;
for i=n+1:n+m
    ich = double(istate(i)) + 3;
    fprintf('%7s%6s%10.2f%14.2f%14.2e%13.2f\n',cch(i-n,:),...
            chstate(ich,:),y(i-n),bl(i),bu(i),clamda(i))
end

```

9.2 Program Results

h02bz example results

Final IP objective value = 97.0000

Varbl	State	Value	Lower Bound	Upper Bound	Lagr Mult
Oatmeal	EQ	4.00	4.00	4.00	3.00
Chicken	LL	0.00	0.00	3.00	24.00
Eggs	LL	0.00	0.00	2.00	13.00
Milk	LL	5.00	5.00	8.00	9.00
Pie	EQ	2.00	2.00	2.00	20.00
Bacon	LL	0.00	0.00	2.00	19.00
L Con	State	Value	Lower Bound	Upper Bound	Lagr Mult
Energy	FR	2080.00	2000.00	1.00e+20	0.00
Protein	FR	64.00	55.00	1.00e+20	0.00
Calcium	FR	1477.00	800.00	1.00e+20	0.00
