NAG Toolbox

nag_mip_iqp_sparse_optstr (h02cg)

1 Purpose

To supply individual optional parameters to nag_mip_iqp_sparse (h02ce).

2 Syntax

```
nag_mip_iqp_sparse_optstr(str)
h02cg(str)
```

3 Description

nag_mip_iqp_sparse_optstr (h02cg) may be used to supply values for optional parameters to nag_mip_iqp_sparse (h02ce). It is only necessary to call nag_mip_iqp_sparse_optstr (h02cg) for those arguments whose values are to be different from their default values. One call to nag_mip_iqp_sparse_optstr (h02cg) sets one argument value.

Each optional parameter is defined by a single character string of up to 72 characters, consisting of one or more items. The items associated with a given option must be separated by spaces, or equal signs [=]. Alphabetic characters may be upper or lower case. The string

```
Print level = 1
```

is an example of a string used to set an optional parameter. For each option the string contains one or more of the following items:

– a mandatory keyword;

– a phrase that qualifies the keyword;

– a number that specifies an integer or double value. Such numbers may be up to 16 contiguous characters in Fortran 77's I, F, E or D formats, terminated by a space if this is not the last item on the line.

Blank strings and comments are ignored. A comment begins with an asterisk (*) and all subsequent characters in the string are regarded as part of the comment.

Normally, each user-specified option is printed as it is defined, on the current advisory message unit (see nag_file_set_unit_advisory (x04ab)), but this printing may be suppressed using the keyword **Nolist**. Thus the statement

```
h02cg('Nolist')
```

suppresses printing of this and subsequent options. Printing will automatically be turned on again after a call to nag_mip_iqp_sparse (h02ce), and may be turned on again at any time using the keyword **List**.

Optional parameter settings are preserved following a call to nag_mip_iqp_sparse (h02ce), and so the keyword **Defaults** is provided to allow you to reset all the optional parameters to their default values by the statement

```
h02cg('Defaults')
```

prior to a subsequent call to nag_mip_iqp_sparse (h02ce).

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in nag_mip_iqp_sparse (h02ce).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **str** – CHARACTER(*)

A single valid option string (as described in Section 3 above and in Section 12 in nag_mip_iqp_sparse (h02ce)).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

None.

6 Error Indicators and Warnings

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

See Section 10 in nag_mip_iqp_sparse (h02ce).

9.1 Program Text

```
function h02cg_example

fprintf('h02cg example results\n\n');

n     = nag_int(7);
m     = nag_int(8);
iobj  = m;
ncolh = n;
big   = 1.e25;

a = [ 1.00  0.15  0.03  0.02  0.02  0.70  0.02   -200 ...
      1.00  0.04  0.05  0.04  0.03  0.75  0.06  -2000 ...
      1.00  0.02  0.08  0.01        0.80  0.08  -2000 ...
      1.00  0.04  0.02  0.02        0.75  0.12  -2000 ...
      1.00  0.02  0.06  0.02  0.01  0.80  0.02  -2000 ...
      1.00  0.01  0.01              0.97  0.01    400 ...
      1.00  0.03                          0.97    400];
ha = nag_int([1 2 3 4 5 6 7 8 ...
              1 2 3 4 5 6 7 8 ...
              1 2 3 4   6 7 8 ...
              1 2 3 4   6 7 8 ...
              1 2 3 4 5 6 7 8 ...
              1 2 3     6 7 8  ...
              1 2         7 8]);
ka = [nag_int(1) 9 17 24 31 39 45 49];
bl = [   0      0    400    100      0     0    0  ...
      2000   -big   -big   -big   -big  1500  250 -big];
bu = [ 200   2500    800    700   1500   big  big  ...
      2000     60    100     40     30   big  300  big];
start = 'C';
names = {'        ', '        ', '        ', '        ', '        '};
```

```
crname = {'...x1...', '...x2...', '...x3...', '...x4...', '...x5...'  ...
          '...x6...', '...x7...', '..row1..', '..row2..', '..row3..'  ...
          '..row4..', '..row5..', '..row6..', '..row7..', '..cost..'};
ns = nag_int(0);
xs = zeros(n+m,1);
intvar = [nag_int(2) 3 4 5 6 7 -1 0 0 0];
istate = zeros(n+m, 1, nag_int_name);
strtgy = nag_int(3);
leniz  = nag_int(100000);
lenz   = nag_int(100000);

% Print Options
h02cg('Nolist');
h02cg('Print level = 0');

[ns, xs, istate, miniz, minz, obj, clamda, ifail] = ...
h02ce( ...
       n, m, iobj, ncolh, @qphx, a, ha, ka, bl, bu, start, names, ...
       crname, ns, xs, intvar, istate, strtgy, leniz, lenz, @monit);

fprintf('Optimal Integer Value is = %20.8e\n',obj);
disp('Components are:');
for j=1:7
  fprintf('x(%2d) = %12.8f\n',j,xs(j));
end

function [hx] = qphx(nstate, ncolh, x)
  hx = zeros(ncolh,1);
  hx(1) = 2*x(1);
  hx(2) = 2*x(2);
  hx(3) = 2*(x(3)+x(4));
  hx(4) = hx(3);
  hx(5) = 2*x(5);
  hx(6) = 2*(x(6)+x(7));
  hx(7) = hx(6);

function [bstval, halt, count] = monit(intfnd,nodes,depth,obj,x,bstval, ...
                                       bstsol,bl,bu,n,halt,count)
                                      halt = false;
  if intfnd == 0
    bstval = -1847510;
  elseif intfnd>count
    fprintf('New integer solution found\n');
    fprintf('  Nodes solved so far: %20d\n', nodes);
    fprintf('  Reached depth:       %20d\n', depth);
    fprintf('  Solution value at current node: %13.5e\n', obj);
    fprintf('  Solution vector at current node:\n');
    fprintf('     %13.5e\n',x);
    fprintf('  Current best function value: %13.5e\n', bstval);
    fprintf('  Current best solution:\n');
    fprintf('     %13.5e\n',bstsol);
    fprintf('  Current lower and upper bounds:\n');
    fprintf('    %13.5e    %13.5e\n', [bl' bu']');
    fprintf('\n');
  end
  count = intfnd;
```

9.2 Program Results

```
     h02cg example results

New integer solution found
  Nodes solved so far:                  272
  Reached depth:                         18
  Solution value at current node:  -1.84752e+06
  Solution vector at current node:
       0.00000e+00
       3.55000e+02
       6.45000e+02
```

```
       1.64000e+02
       4.10000e+02
       2.75000e+02
       1.51000e+02
  Current best function value:  -1.84752e+06
  Current best solution:
       0.00000e+00
       3.55000e+02
       6.45000e+02
       1.64000e+02
       4.10000e+02
       2.75000e+02
       1.51000e+02
  Current lower and upper bounds:
      0.00000e+00      2.00000e+02
      3.55000e+02      3.55000e+02
      4.00000e+02      6.45000e+02
      1.64000e+02      1.64000e+02
      0.00000e+00      1.50000e+03
      0.00000e+00      1.00000e+25
      0.00000e+00      1.00000e+25

Optimal Integer Value is =      -1.84751800e+06
Components are:
x( 1) =   0.00000000
x( 2) = 355.00000000
x( 3) = 645.00000000
x( 4) = 164.00000000
x( 5) = 410.00000000
x( 6) = 275.00000000
x( 7) = 151.00000000
```