GAMBAS: A Program for Saving an Advanced Basis for GAMS
Version 1.0

by

Bruce McCarl
Professor
Texas A&M University
College Station, TX 77843-2124
McCarl@Tamu.edu

GAMSBAS: A Program for Saving an Advanced Basis for GAMS

A program (GAMSBAS) has been written which saves information providing an advanced basis for a GAMS model. This information contains the shadow price, variable levels, and reduced costs, and is saved in a GAMS readable file. The file can, in turn, be included in subsequent GAMS models providing an advanced basis.

General Notes

Use of this procedure is relevant in cases where there are alterations in the data before a SOLVE statement in a large already solved model. The model should usually have saved restart files and the alterations should require rerunning the model from scratch. The general use of the procedure requires restarting GAMS after a solve and executing with a procedure which saves the basis information. The resultant data will be written on the file *.BAS, where "*" is the name of the GAMS input file. This basis then can be included in subsequent runs. An example of this procedure is given below.

The basis file should never be used for one time solution of a problem and rarely for solution of a file without use of restart files. One should only use this procedure with large models when one has to manipulate some of the original data sets equations, or variables before the first solve statement such that the model has to be restarted from scratch. One might also wish to preserve a basis from an alternative run.

The implementation of GAMSBAS causes it to go through several steps. When the procedure is first called GAMS generates the model and sends it out to the solver. In turn, GAMSBAS examines the problem and selects the solver to be used. Ordinarily, the default solver for the problem type (whether linear, nonlinear, or integer) is used. Users may exercise control over this process by using the options file (GAMSBAS.OPT) as described below. In turn, the solver is invoked and then GAMSBAS writes the basis.

During execution the program includes the line $OFFLISTING as the sixth line in the *.BAS file. This suppresses the listing of all but the first five lines of the basis in the file that includes it. Users wishing the full listing should delete this entry.

GAMS constructs a basis using information from the optimal solution. This ordinarily involves the level and marginal value of all variables plus an indicator of whether or not an equation has a shadow price. Degeneracies and alternative optimals complicate this process. GAMSBAS tries to overcome this by inserting EPS to indicate when a variable is basic or nonbasic.

Once the GAMSBAS information has been placed into GAMS the basis may not always be adequate. For example, a model which took over 100,000 iterations to get an initial solution required 1200 iterations to reach optimality when restarted from its GAMSBAS basis. However, this reflected a considerable time saving.

Program Usage

There are three steps involved in using GAMSBAS. The first step involves changing the solver name in the GAMS file. This is done using the command:

OPTION LP=GAMSBAS
or
OPTION NLP=GAMSBAS
or
OPTION IP=GAMSBAS

The solver in this case is named GAMSBAS.

Second, restart the model and generate the basis file. Let's assume that the model name is BLOCKDIA. One would then execute the command GAMS BLOCKDIA with the solve option inserted before the solve command, as is done in Table 1, line 147. In turn, the file BLOCKDIA.BAS is generated. This file is listed in Table 2. Note, this file is just a set of GAMS replacement commands which inserts marginal values for the equations and marginal and level values for the variables (See the chapter on Basis formation in McCarl et al for an explanation of GAMS basis formation).

Third, an include command is entered right before the solve in the model to be restarted and the option selecting GAMSBAS as the solving program is normally eliminated. This is done in Table 3 in lines 147-8 (note the OPTION LP=GAMSBAS is commented out). Use of this procedure results in the model in Table 1 solving in 0 iterations after inclusion of the basis file as opposed to 23 iterations before inclusion of the basis file.

One may find that when a basis from one model is included in another model that the compiler may detect domain errors because the variables are defined over sets with different structures across the two models. One can suppress the domain errors by using the GAMS command $OFFUNI just before inclusion of the basis file and $ONUNI just after.

The OPTION FILE

GAMSBAS internally selects the solver to use. Users may override this choice by the use of the options file. There are keywords allowed in the options file. These are as follows

OPTION Name	Purpose
LP	Gives name of solver for LP problem
NLP	Gives name of solver for NLP problem
MIP	Gives name of solver for MIP problem
DNLP	Gives name of solver for DNLP problem
SOLVERNAME	Gives name of solver for problem to be used

In each case the option name is followed by the name of one of a licensed solvers. If the options file is empty then the default solvers will be used provided it matches the name of a solver GAMS knows about.

The GAMSBAS option file is called GAMSBAS.OPT. An example of a file could look like the following 2 lines

```
LP      OSL
MIP     LAMPS
```

One other important point regarding the option file involves the name of the active solver options file. As seen above the GAMSBAS.OPTION file does not include options commands such as those which should be submitted to MINOS for example. In all cases the program uses the default option filename for the particular solver. Thus if MINOS5 is being used the program looks for the solver option file on MINOS5.OPT.

References

Brooke, A., D. Kendrick, and A. Meeraus. GAMS: A User's Guide. The Scientific Press, South San Francisco, CA, 1988.

McCarl, B.A. “So Your GAMS Model Didn’t Work Right: A Guide to Model Repair.” Texas A&M University, College Station, TX, 1994.

McCarl, B.A., and T.H. Spreen. "Applied Mathematical Programming Using Algebraic Systems." Draft Book, Department of Agricultural Economics, Texas A&M University, College Station, TX, 1996.

Table 1. Example File

```
*     SECTION A       SET DEFINITION

 SET  PRODUCT       TABLES CHAIRSSETS    /TABLES, CHAIRS, DINSETS/
      TYPE          TYPES OF PRODUCT     /FUNCTIONAL ,FANCY/
      RESOURCE      TYPES OF RESOURCES   /SMLLATHE,LRGLATHE,CARVER,LABOR,
                                                                      TOP/
      METHOD        PRODUCTION METHODS   /NORMAL,MAXSML,MAXLRG/
      PLANT         DIFFERENT PLANTS     /PLANT1, PLANT2, PLANT3/
      SUBPRODUCT(PRODUCT)                /TABLES, CHAIRS/;

*     SECTION B       DATA DEFINITION

 PARAMETER  SETCHAIR(TYPE)  CHAIRS CONTAINED IN EACH SET
                           / FUNCTIONAL 4, FANCY 6/
           TABLECOST(TYPE)  TABLECOST  /FUNCTIONAL 80,FANCY 100/;

 TABLE  CHAIRCOST(METHOD,TYPE) CHAIR COST FOR DIFFERENT METHOD
                                FUNCTIONAL      FANCY
            NORMAL                 15            25
            MAXSML                 16            25.7
            MAXLRG                 16.5          26.6 ;

 TABLE TB1(RESOURCE,TYPE,METHOD)   USE OF RESOURCES IN CHAIR PRODUCTION

            FUNCTIONAL.NORMAL   FUNCTIONAL.MAXSML   FUNCTIONAL.MAXLRG
SMLLATHE          0.8               1.30                0.20
LRGLATHE          0.5               0.20                1.30
CARVER            0.4               0.40                0.40
LABOR             1.0               1.05                1.10
 +

             FANCY.NORMAL       FANCY.MAXSML        FANCY.MAXLRG
SMLLATHE         1.2                1.7                  0.50
LRGLATHE         0.7                0.30                 1.50
CARVER           1.0                1.00                 1.00
LABOR            0.8                0.82                 0.84;

 TABLE TB2(RESOURCE,TYPE)     USE OF RESOURCES IN TABLE PRODUCTION

               FUNCTIONAL     FANCY
   LABOR          3            5
   TOP            1            1;

 TABLE TRANSCOST(SUBPRODUCT, PLANT,TYPE) TRANSPORT COST TO PLANT1

             PLANT1.FUNCTIONAL     PLANT2.FUNCTIONAL  PLANT3.FUNCTIONAL
    CHAIRS                                5                7
    TABLES                                                 20
   +
             PLANT1.FANCY     PLANT2.FANCY     PLANT3.FANCY
    CHAIRS                         5                 7
    TABLES                                          20;

 TABLE PRICE(PRODUCT,TYPE) PRICE OF CHAIRS
                  FUNCTIONAL     FANCY
        CHAIRS       82            105
        TABLES      200            300
        DINSETS     600           1100;

 TABLE RESORAVAIL(RESOURCE,PLANT) RESOURCES AVAILABLE
                PLANT1      PLANT2      PLANT3
      TOP         50                      40
      SMLLATHE               140         130
      LRGLATHE                90         100
      CARVER                 120         110
      LABOR       175        125         210;
```

Table 1. Example File (Continued)

```
 TABLE ACTIVITY(PRODUCT,PLANT) TELLS IF A PLANT SELLS A PRODUCT
            PLANT1   PLANT2   PLANT3
  TABLES      1                 1
  CHAIRS               1        1
  DINSETS     1                    ;

*   SECTION     C      MODEL  DEFINITION

 POSITIVE VARIABLES
         MAKECHAIR(PLANT, TYPE,METHOD)  NUMBER OF CHAIRS MADE
         MAKETABLE(PLANT, TYPE)         NUMBER OF TABLES MADE
         TRNSPORT(PLANT,SUBPRODUCT,TYPE) NUMBER OF ITEMS TRANSPORTED
         SELL(PLANT,PRODUCT,TYPE)        NUMBER OF ITEMS SOLD;

 VARIABLES
         NETINCOME    NET REVENUE (PROFIT);
 EQUATIONS
          OBJT         OBJECTIVE FUNCTION ( NET REVENUE )
          RESOUREQ(PLANT,RESOURCE)
          LINKTABLE(TYPE)  OVERALL FIRM TABLE LINKAGE CONSTRAINTS
          LINKCHAIR(TYPE)  OVERALL FIRM CHAIR LINKAGE CONSTRAINTS
          TRNCHAIREQ(PLANT,TYPE) CHAIRS BALANCE FOR A PLANT
          TRNTABLEEQ(PLANT,TYPE) TABLES BALANCE FOR A PLANT;

  OBJT..      NETINCOME =E=
      SUM((TYPE, PRODUCT, PLANT)$ACTIVITY(PRODUCT,PLANT),
                 PRICE(PRODUCT,TYPE) * SELL(PLANT,PRODUCT,TYPE))
    - SUM((PLANT,TYPE)$ACTIVITY("TABLES",PLANT),
             MAKETABLE(PLANT,TYPE)*TABLECOST(TYPE))
    - SUM((PLANT,TYPE,METHOD)$ACTIVITY("CHAIRS",PLANT),
             MAKECHAIR(PLANT,TYPE,METHOD)* CHAIRCOST(METHOD,TYPE))
    - SUM((PLANT,TYPE,SUBPRODUCT)$TRANSCOST(SUBPRODUCT,PLANT,TYPE),
           TRANSCOST(SUBPRODUCT,PLANT,TYPE) * TRNSPORT(PLANT,SUBPRODUCT,

  RESOUREQ(PLANT,RESOURCE)..
        SUM((TYPE,METHOD)$ACTIVITY("CHAIRS",PLANT), TB1(RESOURCE, TYPE,METHOD)
             * MAKECHAIR(PLANT,TYPE,METHOD)) + SUM(TYPE$TB2(RESOURCE, TYPE),
             TB2(RESOURCE,TYPE) * MAKETABLE(PLANT,TYPE))
             =L= RESORAVAIL(RESOURCE,PLANT) ;

  LINKTABLE(TYPE)..
            SUM(PRODUCT$ACTIVITY(PRODUCT,"PLANT1"), SELL("PLANT1", PRODUCT,TYPE))
            =L= MAKETABLE("PLANT1",TYPE) +
                SUM(PLANT$TRANSCOST("TABLES",PLANT,TYPE),
                    TRNSPORT(PLANT,"TABLES",TYPE));

  LINKCHAIR(TYPE)..
             SELL("PLANT1","DINSETS",TYPE) * SETCHAIR(TYPE)
             =L= SUM(PLANT$TRANSCOST("CHAIRS",PLANT,TYPE),
                 TRNSPORT(PLANT,"CHAIRS",TYPE));

  TRNCHAIREQ(PLANT,TYPE)..
             (TRNSPORT(PLANT,"CHAIRS",TYPE) + SELL(PLANT,"CHAIRS",TYPE))
              $TRANSCOST("CHAIRS",PLANT,TYPE)
              =L= SUM(METHOD$ACTIVITY("CHAIRS",PLANT),
                  MAKECHAIR(PLANT,TYPE,METHOD));

  TRNTABLEEQ(PLANT,TYPE)..
             (TRNSPORT(PLANT,"TABLES",TYPE) + SELL(PLANT,"TABLES",TYPE)
              - MAKETABLE(PLANT,TYPE))$TRANSCOST("TABLES",PLANT,TYPE)
              =L= 0 ;

 MODEL Furn /ALL/;

* SECTION D    SOLVE THE PROBLEM
 option lp=gamsbas
 SOLVE Furn USING LP MAXIMIZING NETINCOME;
```

Table 2. Basis File

```
OBJT.m  =    1.00000000000    ;
RESOUREQ.m ("PLANT1","LABOR")  =    44.0000000000    ;
RESOUREQ.m ("PLANT2","SMLLATHE")  =    47.7696526508    ;
RESOUREQ.m ("PLANT2","LRGLATHE")  =    38.8299817185    ;
RESOUREQ.m ("PLANT2","LABOR")  =    19.3692870201    ;
$ OFFLISTING;
RESOUREQ.m ("PLANT3","SMLLATHE")  =    18.4975609756    ;
RESOUREQ.m ("PLANT3","LRGLATHE")  =    12.1853658537    ;
RESOUREQ.m ("PLANT3","CARVER")  =    35.2731707317    ;
RESOUREQ.m ("PLANT3","LABOR")  =    40.0000000000    ;
LINKTABLE.m ("FUNCTIONAL")  =    212.000000000    ;
LINKTABLE.m ("FANCY")  =    320.000000000    ;
LINKCHAIR.m ("FUNCTIONAL")  =    97.0000000000    ;
LINKCHAIR.m ("FANCY")  =    130.000000000    ;
TRNCHAIREQ.m ("PLANT2","FUNCTIONAL")  =    92.0000000000    ;
TRNCHAIREQ.m ("PLANT2","FANCY")  =    125.000000000    ;
TRNCHAIREQ.m ("PLANT3","FUNCTIONAL")  =    90.0000000000    ;
TRNCHAIREQ.m ("PLANT3","FANCY")  =    123.000000000    ;
TRNTABLEEQ.m ("PLANT3","FUNCTIONAL")  =    200.000000000    ;
TRNTABLEEQ.m ("PLANT3","FANCY")  =    300.000000000    ;
MAKECHAIR.l ("PLANT2","FUNCTIONAL","NORMAL")  =    62.2333942718    ;
MAKECHAIR.m ("PLANT2","FUNCTIONAL","MAXSML")  =   -14.2042961609    ;
MAKECHAIR.m ("PLANT2","FUNCTIONAL","MAXLRG")  =   -5.83912248629    ;
MAKECHAIR.l ("PLANT2","FANCY","NORMAL")  =    73.0195003047    ;
MAKECHAIR.m ("PLANT2","FANCY","MAXSML")  =   -9.44021937843    ;
MAKECHAIR.l ("PLANT2","FANCY","MAXLRG")  =    5.17976843388    ;
MAKECHAIR.l ("PLANT3","FUNCTIONAL","NORMAL")  =    35.3658536585    ;
MAKECHAIR.m ("PLANT3","FUNCTIONAL","MAXSML")  =   -8.59317073171    ;
MAKECHAIR.m ("PLANT3","FUNCTIONAL","MAXLRG")  =   -4.14975609756    ;
MAKECHAIR.l ("PLANT3","FANCY","NORMAL")  =    76.8292682927    ;
MAKECHAIR.m ("PLANT3","FANCY","MAXSML")  =   -5.87463414634    ;
MAKECHAIR.l ("PLANT3","FANCY","MAXLRG")  =    19.0243902439    ;
MAKETABLE.l ("PLANT1","FUNCTIONAL")  =    24.3998119826    ;
MAKETABLE.l ("PLANT1","FANCY")  =    20.3601128105    ;
MAKETABLE.m ("PLANT2","FUNCTIONAL")  =   -58.1078610603    ;
MAKETABLE.m ("PLANT2","FANCY")  =   -96.8464351005    ;
MAKETABLE.l ("PLANT3","FANCY")  =    19.4380487805    ;
MAKETABLE.m ("PLANT3", "FUNCTIONAL") = EPS;
TRNSPORT.l ("PLANT2","CHAIRS","FUNCTIONAL")  =    62.2333942718    ;
TRNSPORT.l ("PLANT2","CHAIRS","FANCY")  =    78.1992687386    ;
TRNSPORT.m ("PLANT3","TABLES","FUNCTIONAL")  =   -8.00000000000    ;
TRNSPORT.l ("PLANT3","TABLES","FANCY")  =    8.64870840207    ;
TRNSPORT.l ("PLANT3","CHAIRS","FUNCTIONAL")  =    35.3658536585    ;
TRNSPORT.l ("PLANT3","CHAIRS","FANCY")  =    95.8536585366    ;
SELL.m ("PLANT1","TABLES","FUNCTIONAL")  =   -12.0000000000    ;
SELL.m ("PLANT1","TABLES","FANCY")  =   -20.0000000000    ;
SELL.l ("PLANT1","DINSETS","FUNCTIONAL")  =    24.3998119826    ;
SELL.l ("PLANT1","DINSETS","FANCY")  =    29.0088212125    ;
SELL.m ("PLANT2","CHAIRS","FUNCTIONAL")  =  -10.00000000000    ;
SELL.m ("PLANT2","CHAIRS","FANCY")  =   -20.0000000000    ;
SELL.l ("PLANT3","TABLES","FANCY")  =    10.7893403784    ;
SELL.m ("PLANT3","CHAIRS","FUNCTIONAL")  =   -8.00000000000    ;
SELL.m ("PLANT3","CHAIRS","FANCY")  =   -18.0000000000    ;
NETINCOME.l  =    36206.8788960    ;
```

Table 3. Example with Basis File Included

```
*     SECTION A       SET DEFINITION

 SET  PRODUCT     TABLES CHAIRSSETS    /TABLES, CHAIRS, DINSETS/
      TYPE        TYPES OF PRODUCT     /FUNCTIONAL ,FANCY/
      RESOURCE    TYPES OF RESOURCES  /SMLLATHE,LRGLATHE,CARVER,LABOR, TOP/
      METHOD        PRODUCTION METHODS   /NORMAL,MAXSML,MAXLRG/
      PLANT         DIFFERENT PLANTS     /PLANT1, PLANT2, PLANT3/
      SUBPRODUCT(PRODUCT)                /TABLES, CHAIRS/;

*     SECTION B       DATA DEFINITION

 PARAMETER  SETCHAIR(TYPE)  CHAIRS CONTAINED IN EACH SET
                          / FUNCTIONAL 4, FANCY 6/
           TABLECOST(TYPE)  TABLECOST  /FUNCTIONAL 80,FANCY 100/;

 TABLE  CHAIRCOST(METHOD,TYPE) CHAIR COST FOR DIFFERENT METHOD
                               FUNCTIONAL      FANCY
            NORMAL                15            25
            MAXSML                16            25.7
            MAXLRG                16.5          26.6 ;

 TABLE TB1(RESOURCE,TYPE,METHOD)   USE OF RESOURCES IN CHAIR PRODUCTION

            FUNCTIONAL.NORMAL   FUNCTIONAL.MAXSML   FUNCTIONAL.MAXLRG
SMLLATHE          0.8               1.30                0.20
LRGLATHE          0.5               0.20                1.30
CARVER            0.4               0.40                0.40
LABOR             1.0               1.05                1.10
 +

             FANCY.NORMAL       FANCY.MAXSML        FANCY.MAXLRG
SMLLATHE         1.2                1.7                  0.50
LRGLATHE         0.7                0.30                 1.50
CARVER           1.0                1.00                 1.00
LABOR            0.8                0.82                 0.84;

 TABLE TB2(RESOURCE,TYPE)     USE OF RESOURCES IN TABLE PRODUCTION

                FUNCTIONAL     FANCY
   LABOR          3           5
   TOP            1           1;

 TABLE TRANSCOST(SUBPRODUCT, PLANT,TYPE) TRANSPORT COST TO PLANT1
             PLANT1.FUNCTIONAL     PLANT2.FUNCTIONAL  PLANT3.FUNCTIONAL
    CHAIRS                                5               7
    TABLES                                                20
   +
             PLANT1.FANCY     PLANT2.FANCY     PLANT3.FANCY
    CHAIRS                         5               7
    TABLES                                        20;

 TABLE PRICE(PRODUCT,TYPE) PRICE OF CHAIRS
                   FUNCTIONAL     FANCY
        CHAIRS       82            105
        TABLES      200            300
        DINSETS     600           1100;

 TABLE RESORAVAIL(RESOURCE,PLANT) RESOURCES AVAILABLE
                  PLANT1      PLANT2      PLANT3
      TOP           50                      40
      SMLLATHE                  140        130
      LRGLATHE                   90        100
      CARVER                    120        110
      LABOR         175         125        210;

 TABLE ACTIVITY(PRODUCT,PLANT) TELLS IF A PLANT SELLS A PRODUCT
            PLANT1   PLANT2   PLANT3
```

Table 3. Example with Basis File Included (Continued)

```
  TABLES      1                 1
  CHAIRS              1         1
  DINSETS     1                    ;

*   SECTION     C      MODEL  DEFINITION

 POSITIVE VARIABLES
         MAKECHAIR(PLANT, TYPE,METHOD)  NUMBER OF CHAIRS MADE
         MAKETABLE(PLANT, TYPE)         NUMBER OF TABLES MADE
         TRNSPORT(PLANT,SUBPRODUCT,TYPE) NUMBER OF ITEMS TRANSPORTED
         SELL(PLANT,PRODUCT,TYPE)        NUMBER OF ITEMS SOLD;

 VARIABLES
         NETINCOME    NET REVENUE (PROFIT);
 EQUATIONS
          OBJT         OBJECTIVE FUNCTION ( NET REVENUE )
          RESOUREQ(PLANT,RESOURCE)
          LINKTABLE(TYPE)  OVERALL FIRM TABLE LINKAGE CONSTRAINTS
          LINKCHAIR(TYPE)  OVERALL FIRM CHAIR LINKAGE CONSTRAINTS
          TRNCHAIREQ(PLANT,TYPE) CHAIRS BALANCE FOR A PLANT
          TRNTABLEEQ(PLANT,TYPE) TABLES BALANCE FOR A PLANT;

  OBJT..      NETINCOME =E=
      SUM((TYPE, PRODUCT, PLANT)$ACTIVITY(PRODUCT,PLANT),
                 PRICE(PRODUCT,TYPE) * SELL(PLANT,PRODUCT,TYPE))
    - SUM((PLANT,TYPE)$ACTIVITY("TABLES",PLANT),
             MAKETABLE(PLANT,TYPE)*TABLECOST(TYPE))
    - SUM((PLANT,TYPE,METHOD)$ACTIVITY("CHAIRS",PLANT),
             MAKECHAIR(PLANT,TYPE,METHOD)* CHAIRCOST(METHOD,TYPE))
    - SUM((PLANT,TYPE,SUBPRODUCT)$TRANSCOST(SUBPRODUCT,PLANT,TYPE),
           TRANSCOST(SUBPRODUCT,PLANT,TYPE) * TRNSPORT(PLANT,SUBPRODUCT, TYPE));

  RESOUREQ(PLANT,RESOURCE)..
             SUM((TYPE,METHOD)$ACTIVITY("CHAIRS",PLANT), TB1(RESOURCE,TYPE,METHOD)
             * MAKECHAIR(PLANT,TYPE,METHOD)) + SUM(TYPE$TB2(RESOURCE, TYPE),
             TB2(RESOURCE,TYPE) * MAKETABLE(PLANT,TYPE))
             =L= RESORAVAIL(RESOURCE,PLANT) ;

  LINKTABLE(TYPE)..
            SUM(PRODUCT$ACTIVITY(PRODUCT,"PLANT1"), SELL("PLANT1",PRODUCT,TYPE))
            =L= MAKETABLE("PLANT1",TYPE) +
                SUM(PLANT$TRANSCOST("TABLES",PLANT,TYPE),
                    TRNSPORT(PLANT,"TABLES",TYPE));

  LINKCHAIR(TYPE)..
             SELL("PLANT1","DINSETS",TYPE) * SETCHAIR(TYPE)
             =L= SUM(PLANT$TRANSCOST("CHAIRS",PLANT,TYPE),
                 TRNSPORT(PLANT,"CHAIRS",TYPE));

  TRNCHAIREQ(PLANT,TYPE)..
             (TRNSPORT(PLANT,"CHAIRS",TYPE) + SELL(PLANT,"CHAIRS",TYPE))
              $TRANSCOST("CHAIRS",PLANT,TYPE)
              =L= SUM(METHOD$ACTIVITY("CHAIRS",PLANT),
                  MAKECHAIR(PLANT,TYPE,METHOD));

  TRNTABLEEQ(PLANT,TYPE)..
             (TRNSPORT(PLANT,"TABLES",TYPE) + SELL(PLANT,"TABLES",TYPE)
              - MAKETABLE(PLANT,TYPE))$TRANSCOST("TABLES",PLANT,TYPE)
              =L= 0 ;

 MODEL Furn /ALL/;

* SECTION D    SOLVE THE PROBLEM
* option lp=gamsbas
$INCLUDE  "blockdia.bas"
SOLVE Furn USING LP MAXIMIZING NETINCOME;
```