GAMSCHK USER DOCUMENTATION

Version 1.1

A System for Examining the Structure and Solution
Properties of Linear Programming Problems
Solved using GAMS

by

Bruce A. McCarl
Professor
Department of Agricultural Economics
Texas A&M University

(409) 845-7504 (fax)
mccarl@tamu.edu

June 25,1998

GAMSCHK USER DOCUMENTATION

LIST OF TABLES AND FIGURES

GAMSCHK USER DOCUMENTATION

This document describes procedures designed to aid users who wish to examine empirical GAMS models for possible flaws. The conceptual basis for many of the routines herein is supplied in McCarl and Spreen, and McCarl et.al.

This package of routines is designed for use on any GAMS platform, but for now is implemented on the HP, PC, DEC Alpha, IBM RS6000 and SUN workstations. The function of the specific components of GAMSCHK are to:

(a) List coefficients for user selected equations and/or variables using the DISPLAYCR procedure.

(b) List the characteristics of selected groups of variables and/or equations using MATCHIT.

(c) List the characteristics of equation and variable blocks using BLOCKLIST.

(d) Examine a GAMS model to see whether any variables and equations contain specification errors using ANALYSIS.

(e) Generate schematics depicting the characteristics of coefficients by variable and equation blocks using BLOCKPIC.

(f) Generate a schematic for small GAMS models or portions of larger models depicting the location of coefficients by sign and magnitude using PICTURE.

(g) Reconstruct the reduced cost of variables and the activity within equations after a model solution using POSTOPT.

(h) Help resolving problems with unbounded or infeasible models using NONOPT and ADVISORY.

General Notes on Package Usage

GAMSCHK must replace a solver. This is done using a GAMS option statement of the form:

```
OPTION LP= GAMSCHK;
        or
OPTION NLP=GAMSCHK;
        or
OPTION MIP=GAMSCHK;
```

which replaces either the NLP, LP, or MIP solver with GAMSCHK.[1] In turn, the user will invoke the solver using the statement:

```
SOLVE MODELNAME USING LP MINIMIZING OBJNAME;
```

where MODELNAME is the name used in the GAMS MODEL statement; OBJNAME is the objective function name for the model; and the type of solver that GAMSCHK has replaced which must also be able to solve this type of problem (LP, NLP, or MIP) is identified.

The following are examples of GAMS sequences which can be added to the GAMS file:

```
OPTION NLP=GAMSCHK;
SOLVE TRANSPORT USING NLP MINIMIZING Z;
        or
OPTION LP=GAMSCHK;
SOLVE FEED USING LP MINIMIZING COST;
        or
OPTION MIP=GAMSCHK;
SOLVE RESOURCE USING MIP MAXIMIZING PROFIT;
```

Selecting a Procedure and Providing Input -- the *.GCK File

GAMSCHK requires that the user indicate which procedures are to be employed. This is specified through the use of the *.GCK file where the * refers to the filename from the GAMS execution instruction[2]. The general form of that file is:

```
FIRST PROCEDURE NAME
        ITEM SELECTION INPUT

SECOND PROCEDURE NAME
        ITEM SELECTION INPUT
```

Spaces and capitalization are ignored in this input. For example, a *.GCK file could look like

```
DISPLAYCR
```

---

[1] In all cases, users will be able to replace the LP solver. Replacement of the other solvers depends on the solver licenses owned by the user.

[2] Thus, if the GAMS instructions are in the file called MYMODEL, and GAMS is invoked using the DOS command GAMS MYMODEL, then the GCK file would be called MYMODEL.GCK. If GAMS instructions are on the filename with a period in it then the name up to the period will be used, i.e., the GCK file associated with MYMODEL.IT would be MYMODEL.GCK

```
variables
        SELL(*,*,FANCY)
        maketable
Invariables
        transport(plant2,*,fancy)
Equations
        objT
        notthere
inequations
        resourceq(plant1)
PICTURE
```

The first procedure name in this case is DISPLAYCR and the following 10 lines indicate the items to be selected. Then, we also request PICTURE. Selection entries are treated using several assumptions. In particular:

1) If the *.GCK file is empty then it is assumed that the BLOCKPIC procedure is selected.

2) Spaces maybe freely used in the GCK input file.

3) Upper, lower, or mixed case input is accepted.

4) GAMSCHK recognizes certain words. These words are listed in Appendix A and cannot be used as variable or equation names.

The *.GCK file: General Notes on Item Selection

Some of the procedures permit selection of variables, equations or functions. Specifically, the DISPLAYCR, PICTURE, POSTOPT, and MATCHIT procedures accept input identifying the variables and equations to be utilized. Also NONOPT accepts limited input controlling its function. General observations about the selection requests are

1) Variables can be chosen by entering the word *VARIABLE or VARIABLES* possibly with a modifier, followed by variable selection statements.

2) Variables can also be selected using the *INEQUATION or INEQUATIONS* syntax followed by names of equations. Use of this syntax results in selection of variables with coefficients in the named equations.

3) Equations are selected by entering the keyword, *EQUATION or EQUATIONS* possibly with a modifier, followed by equation selection statements.

4) Equations can also be selected using the *INVARIABLE or INVARIABLES* syntax followed by names of variables. Use of this syntax results in selection of equations in which the named variables have coefficients.

5) Certain item selection modifier keywords can be used depending on procedure. The *INTERSECT* keyword works with procedures *DISPLAYCR* and *POSTOPT*. The *INEQUATION* and *INVARIABLE* keywords work with procedures *DISPLAYCR, PICTURE* and *POSTOPT*. *LISTEQUATION* and *LISTVARIABLE* keywords work with the *MATCHIT* procedure. *INSOLUTION, NOTINSOLUTION, BINDING,* and *NOTBINDING* keywords work with *POSTOPT.* The keywords *VERBOSE* and *IDENTIFY* work with NONOPT.

6) If variable or equation names do not follow the keyword, then usually all variables or equations are assumed selected.

When variables or equations are to be selected after an item selection keyword, a number of input conventions apply. These conventions are:

1) If a variable or equation name is entered without any following parentheses, then all cases for that variable or equation are selected.

2) The selection entries identify specific elements from among the sets over which the variables and equations are defined. In specifying these elements one can use various wild card entries as discussed below or an element name. **Note GAMS set or subset names cannot be used. Set membership information is not available to the GAMSCHK routines.**

3) Wild cards can be used to select items. An "*" will select any item. For example, "B*" will select anything starting with a B. "A?B" will select anything beginning with A, ending with B with one intervening alpha numeric character.

4) When individual elements are specified, you need not enclose them in quotes (").

5) Quotes must be specified to include set item names with spaces, and special characters. In that case wild cards do not work and all input up to the next quote is simply copied.

6) When the selected item has more dimensions than specified, then all later dimensions are handled as if a wild card were specified. For example, when a variable X is defined with reference to 4 sets in the GAMS instructions, but only 3 parameters are specified in the GAMSCHK input, then the request is handled as if all elements of the 4th are desired.

7) When the selected item has less dimensions in GAMS than in the item

selection input, then all additional dimensions are ignored. Thus, when a variable X is defined with reference to 3 sets in GAMS, but 4 parameters are specified in the item selection file, then the 4th specification is ignored.

8) Multiple selection statements can appear on successive lines of the *.GCK file. Output is ordered according to the way items are found in the GAMS file which is determined by the ordering of variables, equations, and set elements in the original GAMS input.

9) Error messages will be generated when an entry cannot be matched to a GAMS element.

10) Examples include

X(*,CLEVELAND)	which indicates that X will be selected for any element of the first set where the element in the second set equals CLEVELAND
X(SEATTLE)	when X is two dimensional selects all cases where the first set element is SEATTLE
X(SEATTLE,CHICAGO,Z)	when X is two dimensional selects the case where the first set element equals SEATTLE, and the second element equals CHICAGO. The third is ignored.
X	all X's will be selected
X(S*, C.O, Z)	when X is three dimensional selects where all X's with first element starting with S, second element beginning with C and ending with O and third element Z will be selected.
*	all variables or equations will be selected
{empty selection set}	all variables or equations will be selected

Procedure Output

In all cases the output generated by the procedure will be written to the *.LST file associated with the GAMS call. Thus, if the file is called MODEL with the *.GCK file (MODEL.GCK), then all output will be on MODEL.LST.

Nonlinear Terms

GAMS models examined with GAMSCHK may involve nonlinear terms. In such cases, GAMSCHK uses the value of the nonlinear term sent forth from GAMS which is an accurate marginal, not total value. GAMS develops this value based on the current level value of the variable. This will either be: a) the starting point selected by GAMS, if the model has not been solved, or b) the current solution value, if the model has been solved. The most accurate portrayals of the coefficients will be generated after the model has been solved through a GAMS SOLVE command before invoking GAMSCHK. Some cases may require a solution and/or the specification of a good starting point before using GAMSCHK. Also, nonlinear terms potentially cause misleading coefficients as those values are local marginal, not global, values determined by the current levels of the variables. Nonlinear terms are marked with *** in the DISPLAYCR, POSTOPT, and NONOPT output.

Entering Comments in the *.GCK File

The *.GCK file has been programmed so that users can enter comments. These comments can take one of two forms. Comments that begin with a hash mark are copied to the output when the program runs. Comments which begin with a question mark are simply overlooked. Thus, one can temporarily comment GAMSCHK selection statements making them inactive by putting in question marks. If multiple procedures are being run or if some sort of output is decided to screen in the computer output then the hash marks can be entered. An example of this appears in Figure 1.

Controlling Page Width in the *.GCK File

When running multiple procedures, in particular the pictures with other procedures, it is often desirable to have some procedures run with wide page widths, but the rest with a narrower page width. The GCK file provides the option to narrow the page width using a PW= command. In particular, what one can do is run GAMS with a large page width, i.e. run GAMS BLOCK pw=200, then insert in the GCK file instructions which narrow that page width for selected procedures. An example of this appears in Figure 1. Users should note that the page width can never be made any wider than the default page width when running with GAMS. Information in excess of the page width will be ignored. Thus, if the model is run under the default status which has a page width of 75 characters then GAMSCHK will reduce the page width down to the maximum page width allowed. Consequently, the pw= command can only be used to narrow the page width from the default page width, not increase it.

Running Multiple Procedures

GAMSCHK can run multiple procedures during one job. This is done by simply stacking the sequence of the commands in the .GCK file. A sequence which generates most of the output that is utilized in the examples is given in Figure 1.

Use of the Procedures

The following section describes the procedures available in GAMSCHK and their input requirements.

DISPLAYCR

Brief Purpose: DISPLAYCR displays all coefficients from the empirical model for a set of user selected equations and variables. All nonzero coefficients under each selected variable or in each selected equation are displayed with the associated variable or equation name and coefficient value. The selection entries may refer to all terms in equations /under variables or only those coefficients at the intersection of the selected variables and equations.

Usage Notes: This option mirrors the GAMS LIMCOL and LIMROW options, but allows the user to select the specific items to be displayed. Partial displays within a variable or equation are also allowed using *INTERSECT*. Use of *VARIABLE* and *EQUATION* keywords followed by selection statements allows one to select variables and equations. Use of the *INVARIABLE* command allows users to select the equations which are associated with a particular variable. For example, if one is having trouble with a particular variable and wants to look at competition in the equations in which it appears, then selecting the variable under the *INVARIABLE* command will display the complete contents of all the equations in which the selected variables have coefficients. Similarly, the *INEQUATION* command will display the complete contents of all variables which fall in a particular equation. Nonlinear terms are marked with ***.
When the keyword *INTERSECT* is found then only the coefficients at the intersection of the specified equations and variables are selected. Use of *INTERSECT* with the *INVARIABLE* syntax results in the named variables and the equations in which they fall being selected. Similarly, use of *INTERSECT* with the *INEQUATION* syntax results in selection of the named equations and the variables which fall in those equations.

Note that when GAMS internal scaling features are employed the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : The keyword DISPLAYCR is entered followed by optional lines of item selection input identifying the variables and equations to be displayed. This file can contain the keywords *VARIABLE, INVARIABLE, EQUATION*, and *INEQUATION*, with each followed by a specification of the items to be selected using the procedure input specification conventions that were described above. The keyword *INTERSECT* can also be used. Several special cases are relevant:

a) If none of the above keywords are found after DISPLAYCR and another procedure name does not follow, then the input is assumed to identify variables.

b) If input is found but the *VARIABLE* or *INEQUATION* keyword cannot be found then no variables are assumed selected.

c) If the *VARIABLE* keyword is entered, but is followed by the end of file or an Appendix A reserved word and *INEQUATION* does not appear, then all variables are assumed selected.

d) If the *EQUATION* or *INVARIABLE* keyword cannot be found, then no equations are assumed selected.

e) If the *EQUATION* keyword is entered, but is followed by the end of the file or a reserved word and the *INVARIABLE* command does not occur, then all equations are assumed selected.

f) The keyword *INVARIABLE* is allowed. It should be followed by variable selection statements. In turn, DISPLAYCR selects all equations which have nonzero entries under the *INVARIABLE* selections.

g) The keyword *INEQUATION* may be used. It should be followed by equation selection statements. In turn, DISPLAYCR selects all variables which have nonzero entries in the *INEQUATION* selections.

h) The keyword *INTERSECT* causes only coefficients at the intersection of the specified equations and variables to be displayed. This occurs for all specifications in this run of DISPLAYCR. One should use DISPLAYCR again if some intersecting and some non-intersecting displays are desired.

i) When *INTERSECT* appears along with *INVARIABLE*, the named variable is selected along with all the equations in which it falls. Similarly, when *INTERSECT* and *INEQUATION* appear then all the named equations and the variables appearing in them are selected.

MATCHIT

Brief Purpose: MATCHIT retrieves the names and characteristics of selected variables and equations. The characteristics reported tell whether the items are nonlinear as well as reporting scaling characteristics and counts of the coefficients. MATCHIT will summarize the items which match a request or list all the items individually.

Usage Notes: The input to MATCHIT can include the keywords *VARIABLE* and *EQUATION* along with those keywords with the prefix *LIST* attached. When the *LIST* prefix is not used, the procedure summarizes the characteristics of all items which match the item requests counting the number of matching items, the number of those items which are nonlinear, the total coefficients under or in those items, the number of positive, negative, and nonlinear coefficients that fall under or in those items. This does not list the names of the individual items which match. If the *LIST* prefix is used (entering *LISTVARIABLE* or *LISTEQUATION*) then the individual matching items are printed in the order in which they are encountered. For each matching item the information tells whether it is nonlinear, how many total coefficients it has, the count of positive, negative, and nonlinear coefficients falling under it, and the minimum and maximum absolute values of coefficients under it (excluding the objective function coefficient).
Note that when GAMS internal scaling features are employed then by default scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : This file contains the keyword MATCHIT, followed by optional item selection input data. The optional input identifies the variables and equations to be displayed. This input can contain the keywords *VARIABLE* or *LISTVARIABLE* followed by a specification of the variables to be selected using the procedure input specification conventions that were described above. This can be followed by the keyword *EQUATION* or *LISTEQUATION* and the specified entries.

Several special cases are relevant:

a) If the procedure name is not followed by any selection input, then a count of all variables and equations appears.

b) If the input is found, but the input does not begin with *VARIABLE, EQUATION, LISTVARIABLE*, or *LISTEQUATION* keywords, then the input is assumed to contain variable names.

c) If the *VARIABLE* keyword is entered, but is not followed by variable selection statements, and *LISTVARIABLE* does not appear, then all variables are assumed selected.

d) If the *EQUATION* or *LISTEQUATION* keyword cannot be found, then no equations are assumed selected.

e) If the *EQUATION* keyword is entered, but is not followed by equation selection statements or a *LISTEQUATION* entry, then all equations are assumed selected.

f) The keyword *LISTVARIABLE* is allowed. It should be followed by variable selection statements. In turn, MATCHIT lists all variables which fall under the request.

g) The keyword *LISTEQUATION* may also be used. It should be followed by equation selection statements. In turn, MATCHIT lists all equations which fall under the request.

h) If none of the above keywords are found, the input is assumed to identify variables.

ANALYSIS

Brief Purpose: Analyzes the structure of all variables and equations. Information is given on errors involving obvious model misspecifications causing redundancy, zero variable values, infeasibility, unboundedness, or obvious constraint relaxations in linear programs. The checks are those identified in Tables 1, 2 and 3.

Usage Notes: The analysis tests given in Tables 1 and 2 are utilized to determine if individual variables or equations in the model possess obvious specification errors. One test, for example, considers whether or not in a maximization problem a variable appears which has a positive return in the objective function, but no coefficients in the constraints indicating an obviously unbounded model. Similarly, information is provided on whether certain equations can never be satisfied. For example, tests examine whether an equality equation appears with a negative right hand side and all positives on the left hand side. Also tests see whether the bounds on variables preclude equation satisfaction or make equations redundant (Table 3). In ANALYSIS these tests are applied to each and every variable and equation. The BLOCKPIC and BLOCKLIST routines utilize the tests on a block by block basis. Thus, the messages will be triggered only if every variable or equation in that block has the same problem. Also interactions between variables and equations are not checked so ANALYSIS only finds flaws contained in individual variables/equations.

Input File: The keyword ANALYSIS is all that is accepted.

BLOCKLIST

Brief Purpose : The BLOCKLIST procedure displays the number and characteristics of the items in each GAMS variable and equation block.

Usage Notes: The characteristic information gives:

1) The variable sign restriction or equation inequality type.

2) The number of variables or equations in this block;

3) The number of variables or equations with at least one nonlinear term in this block.

4) The number of positive coefficients under the variables or in the equations.

5) The number of negative coefficients under the variables or in the equations.

6) The number of nonlinear coefficients under the variables or in the equations.

7) The largest coefficient in absolute value in this block;

8) The smallest coefficient in absolute value in this block. Analysis tests are also performed as discussed under the ANALYSIS procedure.

Note that when GAMS internal scaling features are employed, the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File: No input other than the procedure name is needed.

BLOCKPIC

Brief Purpose: Generates model schematics and scaling information. The schematics depict coefficient signs, total and average number of coefficients within each GAMS equation and variable block.

Usage Notes: These schematics are designed to aid users in identifying flaws in coefficient placement and sign. The summary information on problem scaling characteristics is designed to help users in scaling data. The scaling information is usually reported after any GAMS scaling (using the variablename.scale and equationname.scale features) but before solver scaling. (The user can change whether descaling is done - see the options file). Analysis tests are done using the procedures in Tables 1 and 2.

Note that when GAMS internal scaling features are employed

the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : The keyword BLOCKPIC is all that is recognized.

PICTURE

Brief Purpose: Generates a schematic depicting the location, sign and magnitude of coefficients for selected variables and equations. Users can use this schematic to help identify flaws in coefficient placement, magnitude, or sign. Reports are also generated on the number of individual elements in the pictured portions of each variable and equation.

Usage Notes: This output can be quite large, so PICTURE should only be used for small models or model components.

Note that when GAMS internal scaling features are employed, the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File: Optional input instructions may appear after the PICTURE keyword. This input selects the variables and equations to be included. Only coefficients at the intersection of the selected variables and equations are portrayed. The selected item in the .GCK file can contain the keywords *VARIABLE*, or *INVARIABLE* followed by a specification of the selected variables using the procedure input specification conventions above. This can be followed by the keywords *EQUATION* or *INEQUATION* and the specified entries. Several special cases are also relevant:

a) If the *VARIABLE* or *INEQUATION* keywords cannot be found, then all variables are assumed selected.

b) If the *EQUATION* or *INVARIABLE* keywords cannot be found, then all equations are assumed to selected.

c) If the none of the *VARIABLE*, *INVARIABLE*, *EQUATION*, or *INEQUATION* keywords are found, everything is pictured and all other input is ignored.

d) When the *INVARIABLE* keyword is used, then all equations in which those variables have coefficients are selected along with the named variables.

e) When the *INEQUATION* keyword is used, then all variables which have coefficients in the named equations are selected along with the named equations.

POSTOPT

Brief Purpose: Does post optimality computations. In that capacity POSTOPT either:

a) Reconstructs the reduced cost of variables after a GAMS model solution. Modelers can use this information to discover why certain variables are nonbasic or why certain shadow prices take on particular values, or

b) Reconstructs the usage and supply across an equation after a GAMS model solution. Modelers can use this information to discover why certain variables or slacks take on particular values, as well as to find out where items within equations are produced and/or used.

Usage Notes: POSTOPT uses essentially the same input conventions as does DISPLAYCR. Thus, the usage notes in that selection are also relevant here. In addition:

1) POSTOPT requires a solution has been obtained GAMSCHK will automatically cause a solver to be invoked unless suppressed by the options file;

2) Nonlinear terms may not be accurate in the row sums as their marginal value not their total value is used but GAMS will have adjusted the right-hand sides for their presence; and

3) Attention can be restricted to only certain types of variables or equations. Variables that are *INSOLUTION* (Nonzero or with Zero marginals),

NOTINSOLUTION (zero with a nonzero marginal) can be requested, *BINDING* or *NONBINDING* equations can be focussed on.

Note that when GAMS internal scaling features are employed, the default option is that the unscaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : An optional input file is read in, indicating the specific variables desired using the conventions explained under DISPLAYCR above. In addition:

1) One can enter *INSOLUTION* to restrict attention to variables which are nonzero or have zero marginals.

2) One can enter *NOTINSOLUTION* to restrict attention to zero variables.

3) The above entries restrict alteration in all *VARIABLE* or *INEQUATION* selection statements in a POSTOPT run.

4) One can enter *BINDING* to only consider equations with zero slack. Similarly, *NONBINDING* considers equations with nonzero slack.

5) The above equation specifications restrict all sections by all *EQUATION* or *INVARIABLES* items in a POSTOPT run.

ADVISORY

Brief Purpose: To identify variables which could be unbounded or equations and variable bounds which could cause a model to be infeasible.

Usage Notes: The ADVISORY procedure causes a presolution report on the set of all: a) variables which could be unbounded and/or b) equations and variable bounds which could cause infeasibility. The tests used are summarized in Table 3. This procedure identifies all variables which would need to be bounded as well as all constraints which need artificial variables if one wishes to

diagnose problems in a model. The same output is also generated by NONOPT but the ADVISORY version does not require a solution.

Input file: Just the word ADVISORY

NONOPT

Brief Purpose: To help diagnose unbounded and infeasible models.

Usage Notes: The NONOPT procedure can be used in either an informative mode or with models which terminate as unbounded or infeasible. NONOPT will look through an optimal model reporting all variables which may be potentially unbounded or infeasible and all equations which may be infeasible using the checks explained under the ADVISORY section. Also in an unbounded model NONOPT can report the names of unbounded or infeasible variables or equations as well as either budgeting or row summing them. NONOPT runs after a solution and causes a solve to occur.

Input File: NONOPT may be followed by optional keywords *IDENTIFY* or *VERBOSE*. The *IDENTIFY* keyword causes GAMSCHK to report potential unbounded variables and/or infeasible equations. *VERBOSE* causes full budgets and row summing as done by the POSTOPT procedure on infeasible equations, and/or variables as well as unbounded variables and/or equations. Only the last encountered of the *VERBOSE* or *IDENTIFY* keywords will be obeyed. The details on these options are as follows:

1) If the *IDENTIFY* keyword is used, then the rules in Table 3 are applied to the model. Identify also anticipates that large upper bounds and/or artificial variables may be present. In an optimal condition all variable and equation levels that have exponents greater than the user supplied level filter in the options file (or 6 by default) are identified as items which could be involved with an unbounded model. Similarly, all variables or equations with marginals greater in exponent than the user supplied marginal exponent filter will be identified as items potentially involved with an infeasible model.

2) When the *VERBOSE* keyword is read then all variables and

equations which are listed as nonoptimal or infeasible are treated using the budgeting and row summing aspects of POSTOPT.

3) When no keyword is found and the model solution is not optimal then the nonoptimal equations, infeasible equations and/or nonoptimal variables are automatically listed.

Options File

GAMSCHK accepts an option file controlling solver choice (when needed); descaling; and size of the nonoptimal filters; the number of variable and column blocks selection entries allowed. The file is called GAMSCHK.OPT.

Solver Choice Options

GAMSCHK calls for the solution of the problem when the POSTOPT or NONOPT procedures are used. In doing this, GAMSCHK internally selects the default GAMS solver for a problem class. Users may override this choice using the solver options file. Users may also force or suppress the solution process.

There are 9 solver related keywords allowed in the options file. These are as follows:

OPTION	*Purpose*
LP	Gives name of solver for LP problems
NLP	Gives name of solver for NLP problems
MIP	Gives name of solver for MIP problems
DNLP	Gives name of solver for DNLP problems
SOLVERNAME	Gives name of solver to be used regardless of problem type
NOSOLVE	Suppresses solution of the problem
SOLVE	Forces solution of the problem
DESCALE	Controls treatment of scaling
OPTFILE	Solver options file number

In the first five cases, the option name is followed by the name of one of the licensed solvers. If the options file is empty, then the default solver will be used. If a solver name is given, then that solver will be used provided it matches the name of a solver GAMS recognizes.

When Should I Use *SOLVE* or *NOSOLVE*

Ordinarily GAMSCHK will cause a solver to be used if either the POSTOPT or the NONOPT options are used. However, users can force solutions under other cases or

suppress solutions if desired.

One should only force a solution (using the *SOLVE* option) when one wishes to use the solution information after GAMSCHK is done either to examine the solution output or do post optimality calculations. Forcing a solution will not cause GAMSCHK to have improved representations of nonlinear terms. That will only occur when a *SOLVE* statement is executed before the *SOLVE* statement involving GAMSCHK.

Control of Number of Variable and Row Selections Allowed

The GAMSCHK program uses an upper estimate on the number of variable or equation blocks. In rare circumstances users may wish to override this choice. The options for this are

OPTION	*Purpose*
VARBLOCK	Maximum number of variable blocks allowed
EQUBLOCK	Maximum number of equation blocks allowed

These options are followed by a number, but should not be routinely used.

Scaling

GAMS users may be utilizing internal features which involve scaling through the Modelname.SCALEOPT=1, VariableName.SCALE, and EquationName.SCALE options. GAMSCHK can work with these options to create output which reflects scaled, unscaled or partially unscaled output. In particular, the command DESCALE can be entered with one of three options: *NEVER, ALL,* or *PART*. If you enter *NEVER*, then none of the model output will be descaled. If you enter *ALL*, then all of the model output will be descaled. The third option is to use *PART*. In that case the NONOPT and POSTOPT output will be descaled whereas scaled information will be displayed for PICTURE, BLOCKPIC, BLOCKLIST, MATCHIT and DISPLAYCR. The *PART* option allows investigation of scaling. If you do not enter a *DESCALE* option then all information will be reported as if the *PART* option was chosen.

NONOPT Filters

The NONOPT model in "*IDENTIFY*" mode checks through a model solution to identify large marginals and/or large variable values. The limits on these checks are provided by two options

OPTION	*Purpose*

LEVELFILT	Numerical value of exponent on “unbounded levels”
MARGFILT	Numerical value of exponent on “infeasible marginals”

These options provide upper bounds on the exponents of the absolute values for the levels and marginals. They are followed by an integer which gives the exponent. Thus, entries like

LEVELFILT	7
MARGFILT	7

will cause the reporting of all marginals and levels which are greater in absolute value than 10^7

Example Options File

The GAMSCHK option file is called GAMSCHK.OPT. An example of a file could look like the following 6 lines

LP	OSL
MIP	LAMPS
VARBLOCK	50
SOLVE	
DESCALE	PART
LEVELFILT	4

Solver Options File

One other important aspect regarding the options file involves the use of a problem solver options file when a solver such as MINOS5, OSL, LAMPS etc. is also being used. As seen above the GAMSCHK.OPT does not recognize option commands such as those which would be submitted to the programming model solvers - OSL for example. In all cases GAMSCHK will cause the default option file for the solver to be used when invoking the solver. Thus if MINOS5 and the options file is invoked is being used, MINOS5 options are controlled by the option file MINO5.OPT while GAMSCHK.OPT controls GAMSCHK operation.

Users can change the number of the solver options file being used by using the OPTFILE parameter in the options file. OPTFILE 2 would cause use of solver options file .OP2.

Annotated Examples Illustrating Usage

This section provides examples on the use of the GAMSCHK procedures and their associated output. The procedures were used on the sample model listed in Appendix B.

The GAMS file for this model was called BLOCK and the input file BLOCK.GCK. Relevant portions of the output are provided from the BLOCK.LST file along with a brief discussion.

DISPLAYCR

The DISPLAYCR procedure is invoked by entering DISPLAYCR followed by variable selections. Suppose we wish to examine: 1) the *SELL* variables at PLANT1 for *FANCY* items; 2) the *MAKECHAIR* variables at *PLANT2* for *FUNCTIONAL* items produced using the *NORMAL* process; 3) the variable NOTTHERE; 4) the variables in the *LINKCHAIR* equation and 5) all the equations starting with *LINK* for *FANCY* items. The *.GCK file would then include:

```
displaycr
   variables
     SELL(plant1,*,fan*)
     makec*(pl.2,fu*,NORMAL)
     maket*(pl.2)
     notthere
   inequation
     linkchAIR(fancy)
   equation
     linkt*(fancy)
```

This input shows that one may freely use spacing, may type in lower or uppercase and may use wild cards. The input also contains asterisks which are used to indicate a wildcard so that all set elements are matched. On the other hand, if an individual element is desired, then that particular element name is entered.

The output shows what happens when a nonexistent item is selected (in this case, the "NOTTHERE" equation cannot be found). Note the message as below appears both in the output and on the terminal screen.

```
##  Did not find Requested Variable NOTTHERE
```

The relevant part of the output file BLOCK.LST is shown in Figure 2. Several comments are in order. First, the items are listed in Figure 2 not according to request, but are listed in the order in which they are found. Thus, the SELL variables for TABLES at PLANT1 occur after the MAKECHAIR variables since that is their GAMS model order. Second, not all variables that appear are requested in a *VARIABLE* statement, rather some of them are associated with the *INEQUATION* request.

MATCHIT

The MATCHIT procedure displays the characteristics of selected items. Suppose we wish to show information for: 1) all of the *SELL* variables associated with *TABLES*; 2) all the equations as a group; and 3) all the equations starting with RESOU. The *.GCK file for this is given below:

```
matchit
        listvariables
                sell(*,tables)
        equations
                *
        listequations
                resou*(*,labor)
```

The resultant output appears in Figure 3. There are four SELLTABLE variables which match. The first SELLTABLE variable is defined for the PLANT1, TABLES and FUNCTIONAL set elements. That variable is not nonlinear, has 2 total coefficients, one of which is positive, one is negative, and has coefficients which have a minimum absolute value of one, and a maximum of 200 (the objective function coefficients are not included in the variable maximum and minimum). The aggregate results of the request is summarized showing 4 matching variables with 8 coefficients, 4 of which are positive and 4 are negative. Then the requested equation are listed. In this case, all the RESOURCE equations involving LABOR are listed, indicating none of them are nonlinear also with their coefficient counts, minimum and maximum absolute values. Finally, the request matches are summarized.

ANALYSIS

ANALYSIS checks for obvious structural defects in the model. The example model we have been using has three obvious structural defects (Figure 4).

The variables MAKETABLE for PLANT2 are identified as ones which will equal zero because they utilize resources but do not produce or return anything (see Figure 4). We also get information from the equations. Namely, in the model as specified, the equation RESOURCEQ (PLANT2, TOP) will cause all the variables with nonzero coefficients equal zero because it is a $\leq$ restriction with a zero right-hand-side and all the coefficients in that equation are positive (also see Table 1).

Notice that the analysis information constraint will only appear here and not in the BLOCKLIST or BLOCKPIC analysis where the same items appear and the same techniques were used. This occurs since the block level procedures deal with the entire block, rather than individual equations or variables. In this case, only selected items have these problems, not every item in a block.

BLOCKLIST

The BLOCKLIST procedure generates a list of all the items that appear under each and every variable and equation block in the model.

The input for the BLOCKLIST procedure simply requires the BLOCKLIST keyword in the *.GCK file. The example output is given in Figure 5. In that output, each of the variable blocks (which are the items named in the GAMS variable statement) are listed. The characteristics of these blocks also appear indicating:

a) The variable sign restriction with positive variables represented by the symbol >= 0 indicating they are greater than or equal to zero. The variables which are unrestricted in sign are represented by the symbol < 0 > and the variables which are restricted to be zero or negative are represented by <= 0.

b) The number of variables.

c) The number of nonlinear variables.

d) The coefficient counts.

e) The maximum and minimum absolute values for coefficients in those blocks.

This is followed by equation block output which gives the same basic information with the differences being that the type of restriction is given, the number of equations is counted, and the number positive and negative right-hand-sides are reported.

Information is also given on the analysis of the equations utilizing the rules discussed under the ANALYSIS procedure. Note, however, that this analysis is limited to the total entries in a block and doesn't pertain to individual entries in that block.

BLOCKPIC

The BLOCKPIC procedure is invoked by entering the line:

BLOCKPIC

in the *.GCK file. Output for our Example Model is presented in Figure 6.

Panel A consists of an aggregate picture of the model where each variable and column block is assigned to one column while each equation block is assigned one row and then a coefficient either “+”, “-” or “m” is put in each location to indicate the sign of all matrix entries appearing in that place. Here the “+” indicates they are all positive, the “-” indicates

they are all negative and the “m” indicates they are mixed in sign. In turn, in this picture the row type and sign of the right hand side is entered in the last column and zeroes are entered in that column when there are no right hand sides. The last row in this aggregate picture gives the variable type where “+” means greater than or equal to zero, “-” means less than or equal to zero and “u” means unrestricted in sign being potentially positive or negative, “B” means binary, “I” means integer and “S” indicates other variable types.

Part B gives a schematic containing the coefficient counts for variable and equation block intersections. This shows that the MAKECHAIR block has 12 positive coefficients in the objective function (OBJT), 48 positives in the RESOURCEQ constraint, and 12 negatives in the TRNCHAIREQ constraint. These variables have 60 positive coefficients and 12 negative coefficients encompassing 12 total variables which are greater than or equal to. Simultaneously, OBJT contains 12 positive coefficients under the MAKECHAIR block, 4 positives under the MAKETABLE block, 6 posatives under the TRNSPORT block, 10 negatives under the SELL block, and 1 positive in the NETINCOME block being equal to a zero right-hand-side with a total of 1 equation in this block containing 23 positive and 10 negative coefficients. Similarly, we can see that the TRNTABLEEQ equation has 2 total equations in the block with 4 positives and 2 negatives and that the RESOURCEQ constraint has 11 positive right-hand-sides.

Part C gives the same information except for the average variable and equation. In this particular case the average MAKECHAIR variable has one positive objective function coefficient, four positive coefficients in RESOURCEQ equations, and one negative coefficient in the TRNCHAIREQ equation for an average total of 5 positives and 1 negative. There again, 12 variables are identified as greater than or equal to zero. Similarly, the average RESOURCEQ equation has 5 coefficients and there are 12 equations. Notice that the average in the tables only pertains to report of an individual equation in the average coefficient number per variable in a specific equation block. One can not interpret the 2 coefficients under the MAKETABLE variable as indicating there are on average 2 coefficients in RESOURCEQ equation.

Part D gives scaling information pertinent to the problem giving the maximum/minimum numbers in absolute value in each equation and variable. This, for example, shows for the MAKECHAIR variable, the maximum absolute value in the objective function is 26.6, while the minimum is 15. Similarly, across the entire objective function equation the maximum absolute value number is 1,100 and the minimum is 1. Summaries are also given for the entire variable maximum and minimum. These information can be used in deciding how to scale the model.

Finally, analysis results appear in the same format as for the analysis procedure. Note, in this case, the analysis did not find any problems. So the output indicates that the equations and variables passed all analysis tests.

PICTURE

The PICTURE procedure is invoked by entering the line:

PICTURE

In the *.GCK file, additional item selection commands are optional. The ones used in this case are:

```
PICTURE
    invariables
        t*
```

This results in a portrayal of the submatrix involving all variables whose names start with t. The PICTURE output gives a portrayal of the magnitude and sign of all coefficients within selected elements of the model or the entire model. The output for the whole model (since no selection input is specified) is given in Figure 7. This output is divided into three parts.

The first part gives the coding with which the coefficient signs and magnitudes will be reported. In particular, if there is a coefficient in the model which is greater than positive 1000, then the picture will report that coefficient as a "G". Similarly, if the coefficient is exactly equal to 1 then the picture will report a "C", and if the coefficient is a negative one the picture will report a "3". All positive coefficients will be reported as letters and all negative numbers will be reported as digits.

The second part of the PICTURE output gives an item by item portrayal of the model. All items in the PICTURE are identified by the variable or equation block name and a number. There are 6 variables whose name starts with a T all of which are TRNSPORT variables. These variables fall into 2 LINKTABLE equations, 2 LINKCHAIR equations, 4 TRNCHAIREQ equations, and 2 TRNTABLEEQ equations. The coefficients in the schematic indicate the placement sign and magnitude of the coefficients appearing in those matrix positions. For example, under the second TRNSPORT variable in the OBJT equation we have a E indicating a coefficient between 10 and 100. Simultaneously, in the first LINKCHAIR equation we have a 3 indicating that a negative one is in that spot. The PICTURE allows one to visualize the structure of the individual equations.

The third part of the PICTURE output cross references the variable and equation codes used in the PICTURE with their full GAMS names. The TRANSPORT variables are defined by 1-6 but the dictionary shows the first of those variables is the TRANSPORT(PLANT2, CHAIR,FUNCTIONAL). Similarly, the dictionary of equations translates back from the equation codes to the equation names. In general, if the item has 9 or less elements, then the item name will be listed vertically for variables, horizontally for equations. If, on the other hand, 10 or more elements are identified, then the variable name is listed horizontally for vaariables and vertically for equations.

POSTOPT

The POSTOPT procedure reconstructs the reduced costs on the variables and summarizes the activity in the equations. Suppose we wish to limit our attention to binding equations and nonzero variables. Let us reconstruct the reduced cost on the MAKECHAIR and MAKETABLE variables which are in solution at PLANT2 for the FANCY furniture and we wish examine activity in the binding resource equations at Plant 2 involved with the lathes. Also suppose we run this both in and out of intersection mode. We would specify the following BLOCK.GCK file for input then.

```
POSTOPT
        insolution
        binding
        intersect
        variables
                m*(plant2,fancy)
        equation
                res*(plant2,???lathe)
```

The resultant output appears in Figures 8 and 9. Figure 8 contains the output separated when the INTERSECT keyword is deleted. The budgeting part of the output deals with reconstruction of the reduced costs. Each coefficient under a variable, the associated shadow price and their product are recorded. This shows, for example, that the first MAKECHAIR variable at PLANT2 in solution has a zero reduced cost which arises since the OBJT and RESOURCEQ costs offset the $125 revenue from the TRNCHAIREQ equation. This information allows one to see how the returns for a particular variable are spread among the RESOURCEQ equations and also which equations may be preventing variables from entering into production. One could also investigate why the RESOURCEQ equation took on those particular values by looking at variables in solution which use those resources. The budgeting part of POSTOPT allows investigation of the magnitudes of the shadow prices and reduced costs.

The second part of the output deals with the ways the variables appear in the equations (row summing). In this particular case, note the RESOURCEQ (PLANT2, SML LATHE) equation is summarized where the endowment is 140 and it is used by three MAKECHAIR variables with a shadow price of 47.77. The general format is that the variables appearing in an equation are listed as are the coefficients with which they appear in the equation, their levels and the level times the variable. In turn, one can then look at, how equation resources are utilized and passed from one variable to another.

Figure 9 contains the POSTOPT output when the INTERSECT keyword is present. Note that in this particular case the only coefficients listed under the variables are those for the selected RESOURCEQ rows at PLANT2 that involve the lathe. Similarly in the equation

row sums the only variables that appear are those that start with M at PLANT2 that are FANCY. This shows that in intersection mode, only those coefficients at the intersections of the selected items are listed. In this case, that the reduced costs are not fully accounted for nor is the level of activity in the equations.

ADVISORY

A run of ADVISORY uses an input stream with just the keyword

ADVISORY

This generates output like in Figure 10. Therein, the variables which have positive objective function coefficients and are not bounded above, as well as those which have negative objective function coefficients which are not bounded below are listed as potential unbounded variables. Simultaneously, the potential infeasible equations listed are those which are not satisfied when all variables are set to zero. These are listed as well as the variables are bounded away from zero. This then provides information for one to enter artificial variables and/or large upper bounds on variables to prevent unboundedness or infeasibility.

NONOPT

The NONOPT procedure is designed to deal with non optimal models either before or after they are made feasible by the addition of artificial variables and large upper bounds.

NONOPT runs in three different modes. The first mode is triggered by the keyword IDENTIFY. In this case if the model solution is found to be feasible then the first phase of the output contains identical output as under the ADVISORY mode above. The latter phase of the output consists of an identification of a) variables which have very large marginals (which are reported as potential contributes to an infeasible model and are influenced by artificial variables) and b) items which have very large levels which are felt to contribute to a potential unbounded situation which has been prevented large upper bounds. Suppose that we doctor up our model (Appendix B) so that it is both unbounded and infeasible by making the model unbounded by changing line 140 from a less than or equal to constraint to a greater than or equal to section and simultaneously introduce a minus sign in front of the resource availability parameter in line 122. Further, suppose we also insert lines in the vicinity of line 149 where we upper bound the SELL variables using the command

SELL.UP(PLANT, PRODUCT, TYPE)=1000000000;

and we introduce a new variable definition after line 93 which is

ARTRES(PLANT, RESOURCE) Artificial variables for resource equation;

which appears in the model following line 114 with the term

SUM((PLANT, RESOURCE), 1000000000*ARTRES(PLANT, RESOURCE))

and add an additional term in line 122

+ARTRES(PLANT, RESOURCE).

Thereby we have entered a large upper bound on the SELL variables which are the revenue generating variables and artificial variables for the resource equations. Solving the model then using NONOPT with or without the keyword "*IDENTIFY*" leads to the output shown in Figure 11. The first part is the same output as under the ADVISORY procedure discussed above. However in this case notice the SELL variables are not listed because we have imposed an upper bound. The output shows the model has an optimal solution as the artificials are dragged in for the RESOURCEQ equations and the large upper bounds limit the values of the SELL. However, the values of decision variables and shadow prices are distorted reflecting the presence of these. NONOPT presents a report of the large values which were found. In particular, the SELL variable levels all equal the large upper bounds indicating they were unbounded. The output also reveals the TRNCHAIREQ equation levels (which are equivalent to slack variables in those equations) are quite large indicating they are interrelated to the unbounded cause.

On the infeasibility side the reports show that the MAKECHAIR variables and RESOURCEQ equations all have large marginals which reflect the presence of the artificial and tell what restrictions cause the infeasibility problem as do the RESOURCEQ equations. This shows in this case if one reduced the lower bound to below zero on MAKECHAIR variables or made the right hand side positive on the resource equations then the model would become feasible.

NONOPT can also be run with a true infeasible unbounded model. In that case if we introduce the unboundedness modification from above we get the output in Figure 12 which shows the variables which are identified as being nonoptimal or unbounded at the optimal solution. When we introduce the infeasibility modification, we get the output in Figure 13 which shows the variables whose bounds might be modified to make the model optimal and/or the equations that are listed as infeasible at the optimal solution.

Known Bugs

There are a few bugs that can cause GAMSCHK to report improper outputs or results. A list of the known bugs, their symptoms and a remedy is given below.

Symptom	Cause	Remedy
Zero Shadow Prices in POSTOPT	Old GAMS version or no Prior Solve	1) Make sure the model was solved, 2) if it was, do not suppress solve in option file, or 3) update to most recent GAMS version
Descaling Does Not Work	Old Version of GAMS	Update
GAMS Blows up after GAMSCHK Runs	Old GAMS version	Ignore, *.LST file, results are fine, can be fixed by updating to the most recent version of GAMS
POSTOPT has error in budgets equal to twice objective function coefficient for nonlinear maximizations	Old GAMS MINOS version	Switch to a minimization formulation or update GAMS/MINOS
ROWSUM does not fully account for the value of nonlinear terms in POSTOPT	Value of nonlinear terms sent from GAMS are only a marginal value	None planned. GAMSCHK would need reprogramming
Error message about size of VARBLOCK or EQNBLOCK	exceeded maximum number of blocks	Modify option file, enlarging or eliminating parameters
GAMSCHK won't run	Files are not properly installed	Recheck installation. If still doesn't work report to author
Zero shadow prices when using NOSOLVE	Old version of GAMS solvers or Shadow prices suppressed	Try changing GAMSCOMP.TXT lines 2 or 0 to 12 or 10, if that doesn't work update GAMS.

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. On web page http://agrinet.tamu.edu/mccarl

Table 1. Conditions under which a modeler should be advised of potential difficulty for equations without nonlinear terms.

Type of Constraint	Count of coefficients under a variable of this type with a particular sign						Sign of RHS	Type of PS[a/]	Examples[b/]
	Nonnegative		Nonpositive		Unrestricted				
	+	-	+	-	+	-			
≤	≥ 0 [c/]	0	0	≥ 0	0	0	0	Zero Variables - Case 1	$\sum x \le 0$ [d/], $-\sum y \le 0$, $\sum x - \sum y \le 0$
	≥ 0	0	0	≥ 0	0	0	-	Infeasible -Case 2	$\sum x \le -k$, $-\sum y \le -k$, $\sum x - \sum y \le -k$
	0	≥ 0	≥ 0	0	0	0	+ or 0	Redundant -Case 3	$-\sum x \le +k$, $\sum y \le +k$, $-\sum x + \sum y \le k$
=	≥ 0	0	0	≥ 0	0	0	0	Zero Variables - Case 1	$\sum x = 0$, $-\sum y = 0$, $\sum x - \sum y = 0$
	0	≥ 0	≥ 0	0	0	0	0	Zero Variables - Case 1	$-\sum x = 0$, $\sum y = 0$, $-\sum x + \sum y = 0$
	≥ 0	0	0	≥ 0	0	0	-	Infeasible -Case 2	$\sum x - \sum y = -k$
	0	≥ 0	≥ 0	0	0	0	+	Infeasible -Case 2	$-\sum x + \sum y = k$
	0	0	0	0	≥ 0 [e/]	≥ 0 [e/]	0	Zero Variable - Case 1	$z=0$, $-z=0$
≥	0	≥ 0	≥ 0	0	0	0	0	Zero Variables - Case 1	$-\sum x \ge 0$, $\sum y \ge 0$, $-\sum x + \sum y \ge 0$
	0	≥ 0	≥ 0	0	0	0	0 or +	Infeasible -Case 2	$-\sum x \ge k$, $\sum y \ge k$, $-\sum x + \sum y \ge k$
	≥ 0	0	0	≥ 0	0	0	- or 0	Redundant -Case 3	$\sum x \ge -k$,$-\sum y \ge -k$, $\sum x - \sum y \ge -k$

a/ The PS cases indicate, because the variables in this equation follow this pattern, that:
1. The variables appearing with nonzeros in this equation are forced to equal zero.
2. This equation can never be satisfied and is obviously infeasible.
3. This equation is redundant. The nonnegativity conditions are a stronger restriction.

b/ In the examples x denotes indexed non-negative variables, y indexed non-positive variables, and z a single unrestricted variable.

c/ Here and in the cases below at least one nonzero must occur

d/ These entries give examples of the problem covered by each warning. Namely, in the first case examining only the nonnegative variables suppose all those variables have signs ≥ 0 but the right-hand-side is zero. Thus, we have $X \ge 0$ and $X \le 0$ which implies $X = 0$. A warning is generated in that case.

e/ Only one coefficient is allowed.

Table 2. Conditions under which a modeler should be warned about variables in a maximization problem.

Type of Variable	Objective function coefficient sign	Number of a_{ij}'s of a sign in						PS[a/]	Examples
		≥ rows		= rows		in ≤ rows			
		+	-	+	-	+	-		
Nonnegative	+	≥ 0	0	0	0	0	≥ 0	Unbounded Variable case 1	max x[b/] , x + DQ ≥ a -x + EQ≤ b
	-	0	≥ 0	0	0	≥ 0	0	Zero optimal solution case 2	max -x -x + DQ ≥ a x + EQ ≤ b
	0	≥ 0	0	0	0	0	≥ 0	Variable Relaxes constraint case 3	max ox x +DQ ≥ a -x + DQ ≤ b
	0	≥ 0	0	≥ 0[c]	≥ 0[c]	0	≥ 0	Variable Relaxes constraint case 4	max ox x + DQ ≥a x+ FQ = g -x + EQ ≤ b
Nonpositive	-	0	≥ 0	0	0	≥ 0	0	Unbounded Variable case 1	max -y -y + DQ ≥ a y + EQ ≤ b
	+	≥ 0	0	0	0	0	≥ 0	Zero optimal solution case 2	max y[b/] , y + DQ ≥ a -y + EQ≤ b
	0	0	≥ 0	0	0	≥ 0	0	Variable Relaxes constraint case 3	max 0x -y + DQ ≥ a y + EQ ≤ b
	0	0	≥ 0	≥ 0[c]	≥ 0[c]	≥ 0	0	Variable Relaxes constraint case 4	max ox -y + DQ ≥a y+FQ = g y + EQ ≤ b
Unrestricted	+/-	0	0	0	0	0	0	Unbounded Variable case 1	max ± z

a/ PS cases are: The variables which satisfy this condition are:

1) Unbounded as they contribute to the objective function while satisfying the constraints.
2) Obviously zero since they consume constraint resources and have a cost in the objective function.
3) Warning this variable relaxes all constraints in which it appears
4) Warning this variable relaxes all the equality constraints in which it appears in one direction

b/ Here x(y) has a positive objective term and can be increased without ever violating any constraints so x(y) is unbounded.

c/ Only one coefficient can be present in the equality rows

Table 3. Conditions When Model Elements Could be Unbounded or Infeasible

Conditions for Potential Unbounded Variables -- Presence of Bounds

Types of Variables	Sign of Objective in Max Problem	Upper	Lower
≥ 0[a]	+	None	---[c]
≤ 0	-	---	None
Unrestricted	+	None	---
Unrestricted	---	---	None

Conditions for Potential Infeasibility Caused by Bounds on Variables

	Existence of Bounds	
Types of Variables	Lower	Upper
$\geq$ [b]0	+	---
≤ 0	---	---
Unrestricted	+	---
Unrestricted	---	---

Conditions for Potential Infeasibility in Equations

Types of Equations	RHS
$\leq$[d]	-
$\geq$	+
=	+ or -

[a] If a non negative variable has a positive objective function coefficient without an upper bound, then the variable could be unbounded.

[b] If a nonnegative variable has a positive lower bound then it could cause infeasibility.

[c] Any reasonable value can exist for this item

[d] When a less than or equal equation is present it may not be able to be satisfied if it has a negative RHS.

Table 4. Conditions for Potential Infeasibility or Redundancy in Equations Based on Bounds on Variables

	TYPE OF CONSTRAINT		PS
	≤ b	≥ b	
SUM OF THE SMALLEST VALUE[a]	> b	---	INFEASIBLE
	---	> b	REDUNDANT
SUM OF THE LARGEST VALUE[b]	---	< b	INFEASIBILE
	< b	---	REDUNDANT

Note:

a. Suppose X_j is bounded as follows, LB_j (lower bound) $\leq X_j \leq UB_j$ (upper bound), and we have the sum which is either >b or <b, then

S= will be the smallest value which could happen in that sum. If the constraint is ≤b, then if S>b, we know that this constraint will never be satisfied. In the constraint is ≥b, then if S>b, we know that this constraint will not limit any possible X value. Hence, it is redundant.

b. Suppose X_j is bounded as follows, LB_j (lower bound) $\leq X_j \leq UB_j$ (upper bound), and we have the sum which is either >b or <b, then

L= will be the largest value which could happen in that sum. If the constraint is ≤b, if L<b, we know that this constraint will not limit any possible X value. Hence, it is redundant. In the constraint is ≥b, then if L<b, we know that this constraint will never be satisfied.

c.Thanks to Paul Preckel for bringing these tests to the authors' attention.

Figure 1. Sample GCK file Illustrating Multiple Runs

```
pw=72
?this line will be ignored
#generate output for figure 2
displaycr
   variables
     sell(plant1,*,fan*)
     makec*(pl.2,fu*,normal)
     maket*(pl.2)
      notthere
   inequation
     linkchair(fancy)
   equation
     linkt*(fancy)
matchit
  listvariables
     sell(*,tables)
   equations
   *
   listequations
      resou*(*,labor)
analysis
blocklist
?remember to use this must have run gams with at least a 200 pw
pw=200
blockpic
picture
      invariables
            t*
pw=72
postopt
   insolution
   binding
   variables
     m*(plant2,fancy)
   equation
     res*(plant2,???lathe)
postopt
   intersect
   insolution
   binding
   variables
     m*(plant2,fancy)
   equation
     res*(plant2,???lathe)
advisory
nonopt
  identify
```

Figure 2. DISPLAYCR Output

```
----#### Executing DISPLAYCR
    ##  Did not find Requested Equation NOTTHERE
       Remember you must use set elements, not set names to select
----###  DISPLAYING VARIABLES

----## VAR MAKECHAIR
##  MAKECHAIR(PLANT2,FUNCTIONAL,NORMAL)
OBJT                                          15.000
RESOURCEQ(PLANT2,SMLLATHE)                   0.80000
RESOURCEQ(PLANT2,LRGLATHE)                   0.50000
RESOURCEQ(PLANT2,CARVER)                     0.40000
RESOURCEQ(PLANT2,LABOR)                       1.0000
TRNCHAIREQ(PLANT2,FUNCTIONAL)                -1.0000
  UPPER BOUND                                 5000.0

----## VAR MAKETABLE
##  MAKETABLE(PLANT2,FUNCTIONAL)
RESOURCEQ(PLANT2,LABOR)                       3.0000
RESOURCEQ(PLANT2,TOP)                         1.0000

##  MAKETABLE(PLANT2,FANCY)
RESOURCEQ(PLANT2,LABOR)                       5.0000
RESOURCEQ(PLANT2,TOP)                         1.0000

----## VAR TRANSPORT
##  TRANSPORT(PLANT2,CHAIRS,FANCY)
OBJT                                          5.0000
LINKCHAIR(FANCY)                             -1.0000
TRNCHAIREQ(PLANT2,FANCY)                      1.0000

##  TRANSPORT(PLANT3,CHAIRS,FANCY)
OBJT                                          7.0000
LINKCHAIR(FANCY)                             -1.0000
TRNCHAIREQ(PLANT3,FANCY)                      1.0000

----## VAR SELL
##  SELL(PLANT1,TABLES,FANCY)
OBJT                                         -300.00
LINKTABLE(FANCY)                              1.0000

##  SELL(PLANT1,DINSETS,FANCY)
OBJT                                         -1100.0
LINKTABLE(FANCY)                              1.0000
LINKCHAIR(FANCY)                              6.0000

----###    DISPLAYING EQUATIONS
----## EQU LINKTABLE

##  LINKTABLE(FANCY)
MAKETABLE(PLANT1,FANCY)                      -1.0000
TRANSPORT(PLANT3,TABLES,FANCY)               -1.0000
SELL(PLANT1,TABLES,FANCY)                     1.0000
SELL(PLANT1,DINSETS,FANCY)                    1.0000
        =L=                                 0.00000E+00
```

Figure 3. MATCHIT Output

```
----#### Executing MATCHIT

Note Max and Min do not include Obj row coef
                                          Is  Tot Pos Neg Nln  Minimum  Maximum
 ----### Requested Variables              Non Cof Cof Cof Cof   Absolute  Absolute


----## VAR SELL

SELL(PLANT1,TABLES,                       0   2      1      1    0             1.000    200
     FUNCTIONAL)
SELL(PLANT1,TABLES,FANCY)                 0   2      1       1   0             1.000    300
SELL(PLANT3,TABLES,                       0   2      1       1   0             1.000    200
     FUNCTIONAL)
SELL(PLANT3,TABLES,FANCY)                 0   2       1       1  0             1.000    300


                          Numb   Numb  Total  Pos   Neg  Nonln
----### Variable Request   Varia  Nonln  Coef  Coef  Coef  Coef

SELL(*,TABLES)              4      0     8     4      4     0

Note Max and Min do not include Rhs and Obj var coef

                            Is  Tot Pos Neg Nln  Minimum   Maximum
 ----### Requested Equations  Non Cof Cof Cof Cof  Absolute  Absolute


----## EQU RESOURCEQ

RESOURCEQ(PLANT1,LABOR)         0   3   3   0   0    1.000     5.000
RESOURCEQ(PLANT2,LABOR)         0   9   9   0   0    .8000     5.000
RESOURCEQ(PLANT3,LABOR)         0   9   9   0   0    .8000     5.000


                             Numb  Numb  Total   Pos   Neg   Nonln
 ----### Equation Request      Equat  Nonln  Coef  Coef  Coef   Coef

All Equations                  23     0     155   113    42     0
RESOU*(*,LABOR)                 3     0      21    21     0     0
```

Figure 4. ANALYSIS Output

```
 ----#### Executing ANALYSIS


   ----### Analysis of Variables ( nonlinear terms at current point)


 **** Warning  These variables will equal zero
               because they have a zero lower bound
               an undesirable object function coefficient
               all 0 or - coefficients in the =G= rows
               all 0 or + coefficients in the =L= rows
               and no coefficients in the =E= rows

 ##  MAKETABLE(PLANT2,FUNCTIONAL)
     MAKETABLE(PLANT2,FANCY)

 ----### Analysis of Equations ( nonlinear terms at current point)


 **** Warning  This =L= constraint causes all
               variables in it to have a zero solution value
               since the nonnegative variables present
                         have only 0 or + coefficients
                     the nonpositive variables present
                         have only 0 or - coefficients
                     the unrestricted variables
                         have only zero coefficients
                     and the rhs is zero.

 ##  RESOURCEQ(PLANT2,TOP)
```

Figure 5. BLOCKLIST Output

```
 ----#### Executing BLOCKLIST

 ----### List of Variable Block Characteristics

 Note Max and Min do not include Objective Row

 Variable   Sign  Numb Numb Pos  Neg  Nonl  Maximum   Minimum
   Block    Res   Vars Nonl Coef Coef Coef  Absolute  Absolute
 MAKECHAIR  >=0    12    0   60   12    0    1.700     .2000
 MAKETABLE  >=0     6    0   16    4    0    5.000     1.000
 TRANSPORT  >=0     6    0   12    6    0    1.000     1.000
 SELL       >=0    10    0   12   10    0    6.000     1.000
 NETINCOME  <0>     1    0    1    0    0    .0000E+00 .0000E+00


 ----### List of Equation Block Characteristics

 Note Max and Min do not include RHS and Objective variable

 Equation   Type  Numb Numb Pos  Neg  Nonl Pos  Neg  Maximum  Minimum
   Block    Res   Eqns Nonl Coef Coef Coef RHS  RHS  Absolute  Absolute
 OBJT       =E=     1    0   23   10    0    0    0   1100.     5.000
 RESOURCEQ  =L=    12    0   60    0    0   11    0   5.000     .2000
 LINKTABLE  =L=     2    0    4    4    0    0    0   1.000     1.000
 LINKCHAIR  =L=     2    0    2    4    0    0    0   6.000     1.000
 TRNCHAIREQ =L=     4    0    8   12    0    0    0   1.000     1.000
 TRNTABLEEQ =L=     2    0    4    2    0    0    0   1.000     1.000

----### Analysis of Variables ( nonlinear terms at current point)

    ### The variables pass all analysis tests

----### Analysis of Equations ( nonlinear terms at current point)

       ### The equations pass all analysis tests
```

Figure 6. BLOCKPIC Output

Panel A. Aggregate Block Picture

```
              |                 |
              |  M  M  T     N  |
              |  A  A  R     E  |
              |  K  K  A     T  |
              |  E  E  N     I  |
              |  C  T  S     N  |
              |  H  A  P  S  C  |
              |  A  B  O  E  O  |     R
              |  I  L  R  L  M  |     H
              |  R  E  T  L  E  |     S
 -------------------------------------
 OBJT         |  +  +  +  -  +  |  E  0
 RESOURCEQ    |  +  +           |  L  +
 LINKTABLE    |     -  -  +     |  L  0
 LINKCHAIR    |        -  +     |  L  0
 TRNCHAIREQ   |  -     +  +     |  L  0
 TRNTABLEEQ   |     -  +  +     |  L  0
 -------------------------------------
 Variable Typ |  +  +  +  +  u
```

Panel B. Picture Giving Number of Coefficients by Block

```
              |                                |             C     #
              |    M     M     T           N  |             o
              |    A     A     R           E  |             e     o
              |    K     K     A           T  |             f     f
              |    E     E     N           I  |             f
              |    C     T     S           N  |
              |    H     A     P     S     C  |             C     E
              |    A     B     O     E     O  |      R      n     q
              |    I     L     R     L     M  |      H      t     n
              |    R     E     T     L     E  |      S      s     s
 ---------------------------------------------------------------------
 OBJT         |   12+    4+    6+          1+ | E          23+    1
              |                      10-      |             10-
 RESOURCEQ    |   48+   12+                   | L   11+    60+   12
              |                               |
 LINKTABLE    |                       4+      | L           4+    2
              |          2-    2-             |             4-
 LINKCHAIR    |                       2+      | L           2+    2
              |                4-             |             4-
 TRNCHAIREQ   |                4+    4+       | L           8+    4
              |   12-                         |            12-
 TRNTABLEEQ   |                2+    2+       | L           4+    2
              |          2-                   |             2-
 ---------------------------------------------------------------------
 Coeff Cnts   |   60+   16+   12+   12+    1+ |     11+  101+
              |   12-    4-    6-   10-       |           32-
 # of  Vars   |   12     6     6    10     1  |
 Variable Typ |   >=0   >=0   >=0   >=0   <0>
```

Figure 6. BLOCKPIC Output (continued)

Panel C. Picture Giving Average Number of Coefficients by Column Block

	MAKECHAIR	MAKETABLE	TRANSPORT	SELL	NETINCOME	RHS	Cfs Per Eqn	# of Vars
OBJT	1+	0.667+	1+		1+	E	23+	1
				1-			10-	
RESOURCEQ	4+	2+				L	11+ 5+	12
LINKTABLE			0.4+			L	2+	2
		0.333-	0.333-				2-	
LINKCHAIR			0.2+			L	1+	2
			0.667-				2-	
TRNCHAIREQ			0.667+	0.4+		L	2+	4
	1-						3-	
TRNTABLEEQ			0.333+	0.2+		L	2+	2
		0.333-					1-	
Cfs PerVar	5+	2.667+	2+	1.2+	1+			
	1-	0.667-	1-	1-				
# of Vars	12	6	6	10	1			
Var Type	>=0	>=0	>=0	>=0	<0>			

Panel D. Scaling Data - Maximum & Minimum Coefficients by Block

		MAKECHAIR	MAKETABLE	TRANSPORT	SELL	NETINCOME	RHS Max	Equ Max
OBJT	Max	26.6	100	20	1100	1		1100
	Min	15	80	5	82	1		1
RESOURCEQ	Max	1.7	5				210	5
	Min	0.2	1				40	0.2
LINKTABLE	Max		1	1	1			1
	Min		1	1	1			1
LINKCHAIR	Max			1	6			6
	Min			1	4			1
TRNCHAIREQ	Max	1		1	1			1
	Min	1		1	1			1
TRNTABLEEQ	Max		1	1	1			1
	Min		1	1	1			1
Total Var	Max	26.6	100	20	1100	1	210	
	Min	0.2	1	1	1	1	40	

Figure 7. PICTURE Output

PANEL A. COEFFICIENT CODES

LOWER BOUND (INCLUSIVE)	CODE	UPPER BOUND (LESS THAN)
1000.00000	G	+INFINITY
100.00000	F	1000.00000
10.00000	E	100.00000
1.00000	D	10.00000
1.00000	C	1.00000
.50000	B	1.00000
.00000	A	.50000
.00000	0	.00000
-.50000	1	.00000
-1.00000	2	-.50000
-1.00000	3	-1.00000
-10.00000	4	-1.00000
-100.00000	5	-10.00000
-1000.00000	6	-100.00000
-INFINITY	7	-1000.00000

Panel B. Picture

```
            |                    R
            | T T T T T T        H
            | R R R R R R        S      P      N
            | A A A A A A               O      E      R
            | N N N N N N        C      S      G      O
            | S S S S S S        O      I A    A A    W
            | P P P P P P        E      T I    T I    C
            | O O O O O O        F      I J      J    N
            | R R R R R R        F      V ,    V ,    T
            | T T T T T T        S      E S    E S    S
            |
            | 1 2 3 4 5 6
            ------------
      OBJT 1| D E E D D 6    = 0      1 1      1    1 2
 LINKTABLE 1|     3          < 0        0      1      1
 LINKTABLE 2|       3        < 0        0      1      1
 LINKCHAIR 1| 3       3      < 0        0      2      2
 LINKCHAIR 2|   3       3    < 0        0      2      2
TRNCHAIREQ 1| C            < 0        1      0      1
TRNCHAIREQ 2|   C          < 0        1      0      1
TRNCHAIREQ 3|         C    < 0        1      0      1
TRNCHAIREQ 4|           C  < 0        1      0      1
TRNTABLEEQ 1|     C        < 0        1      0      1
TRNTABLEEQ 2|       C      < 0        1      0      1
            ------------
 LOWER BND  | 0 0 0 0 0 0
 UPPER BND  | + + + + + +
            ------------
 POSITIVE   | 3   3   3
 COLUMN CTS |   3   3   2
 NEGATIVE   | 1   1   1
 COLUMN CTS |   1   1   2
 COLUMN     | 4   4   4
 COUNTS     |   4   4   4
            ------------
```

Figure 7. PICTURE Output (Continued)

Panel C. Dictionary of Variables and Equations

```
----### Dictionary of Variables

TRANSPORT    1: TRANSPORT(PLANT2,CHAIRS,FUNCTIONAL)
TRANSPORT    2: TRANSPORT(PLANT2,CHAIRS,FANCY)
TRANSPORT    3: TRANSPORT(PLANT3,TABLES,FUNCTIONAL)
TRANSPORT    4: TRANSPORT(PLANT3,TABLES,FANCY)
TRANSPORT    5: TRANSPORT(PLANT3,CHAIRS,FUNCTIONAL)
TRANSPORT    6: TRANSPORT(PLANT3,CHAIRS,FANCY)


----### Dictionary of Equations

OBJT         1: OBJT
LINKTABLE    1: LINKTABLE(FUNCTIONAL)
LINKTABLE    2: LINKTABLE(FANCY)
LINKCHAIR    1: LINKCHAIR(FUNCTIONAL)
LINKCHAIR    2: LINKCHAIR(FANCY)
TRNCHAIREQ   1: TRNCHAIREQ(PLANT2,FUNCTIONAL)
TRNCHAIREQ   2: TRNCHAIREQ(PLANT2,FANCY)
TRNCHAIREQ   3: TRNCHAIREQ(PLANT3,FUNCTIONAL)
TRNCHAIREQ   4: TRNCHAIREQ(PLANT3,FANCY)
TRNTABLEEQ   1: TRNTABLEEQ(PLANT3,FUNCTIONAL)
TRNTABLEEQ   2: TRNTABLEEQ(PLANT3,FANCY)
```

Figure 8. POSTOPT Output - Normal Mode

```
----#### Executing POSTOPT

----###  BUDGETING VARIABLES
           PRINTING VARS IN SOLUTION
           PRINTING BINDING EQNS

----## VAR MAKECHAIR

##  MAKECHAIR(PLANT2,FANCY,NORMAL)
  SOLUTION VALUE                                73.0195
  UPPER BOUND                                   5000.00

  EQN                                  Aij         Ui       Aij*Ui
OBJT                                 25.000      1.0000      25.000
RESOURCEQ(PLANT2,SMLLATHE)           1.2000      47.770      57.324
RESOURCEQ(PLANT2,LRGLATHE)           .70000      38.830      27.181
RESOURCEQ(PLANT2,LABOR)              .80000      19.369      15.495
TRNCHAIREQ(PLANT2,FANCY)            -1.0000      125.00     -125.00
TRUE REDUCED COST                                            .00000E+00


##  MAKECHAIR(PLANT2,FANCY,MAXLRG)
  SOLUTION VALUE                                5.17977
  UPPER BOUND                                   5000.00

  EQN                                  Aij         Ui       Aij*Ui
OBJT                                 26.600      1.0000      26.600
RESOURCEQ(PLANT2,SMLLATHE)           .50000      47.770      23.885
RESOURCEQ(PLANT2,LRGLATHE)           1.5000      38.830      58.245
RESOURCEQ(PLANT2,LABOR)              .84000      19.369      16.270
TRNCHAIREQ(PLANT2,FANCY)            -1.0000      125.00     -125.00
TRUE REDUCED COST                                            .00000E+00


 ----###   ROW SUMMING EQUATIONS
           PRINTING VARS IN SOLUTION
           PRINTING BINDING EQNS



----## EQU RESOURCEQ


##  RESOURCEQ(PLANT2,SMLLATHE)

  VAR                                  Aij         Xj        Aij*Xj
MAKECHAIR(PLANT2,FUNCTIONAL,         .80000      62.233      49.787
         NORMAL)
MAKECHAIR(PLANT2,FANCY,NORMAL)       1.2000      73.020      87.623
MAKECHAIR(PLANT2,FANCY,MAXLRG)       .50000      5.1798      2.5899
   =L=                                                        =L=
RHS COEFF                                                   140.00

SLACK EQUALS                                                .00000E+00
SHADOW PRICE                                                47.770

##  RESOURCEQ(PLANT2,LRGLATHE)

  VAR                                  Aij         Xj        Aij*Xj
MAKECHAIR(PLANT2,FUNCTIONAL,         .50000      62.233      31.117
         NORMAL)
MAKECHAIR(PLANT2,FANCY,NORMAL)       .70000      73.020      51.114
MAKECHAIR(PLANT2,FANCY,MAXLRG)       1.5000      5.1798      7.7697
   =L=                                                        =L=
RHS COEFF                                                   90.000

SLACK EQUALS                                                .00000E+00
SHADOW PRICE                                                38.830
```

Figure 9. POSTOPT Output - Intersection Mode

```
----#### Executing POSTOPT


----###  BUDGETING VARIABLES
           INTERSECTION MODE
           PRINTING VARS IN SOLUTION
           PRINTING BINDING EQNS


----## VAR MAKECHAIR


##  MAKECHAIR(PLANT2,FANCY,NORMAL)
  SOLUTION VALUE                                73.0195
  UPPER BOUND                                   5000.00

  EQN                                  Aij         Ui       Aij*Ui
RESOURCEQ(PLANT2,SMLLATHE)           1.2000      47.770      57.324
RESOURCEQ(PLANT2,LRGLATHE)           .70000      38.830      27.181
SUM OF THESE TERMS                                           84.505
TRUE REDUCED COST                                            .00000E+00


##  MAKECHAIR(PLANT2,FANCY,MAXLRG)
  SOLUTION VALUE                                5.17977
  UPPER BOUND                                   5000.00

  EQN                                  Aij         Ui       Aij*Ui
RESOURCEQ(PLANT2,SMLLATHE)           .50000      47.770      23.885
RESOURCEQ(PLANT2,LRGLATHE)           1.5000      38.830      58.245
SUM OF THESE TERMS                                           82.130
TRUE REDUCED COST                                            .00000E+00



 ----###   ROW SUMMING EQUATIONS
           INTERSECTION MODE
           PRINTING VARS IN SOLUTION
           PRINTING BINDING EQNS



----## EQU RESOURCEQ


##  RESOURCEQ(PLANT2,SMLLATHE)

  VAR                                   Aij         Xj       Aij*Xj
MAKECHAIR(PLANT2,FANCY,NORMAL)       1.2000      73.020      87.623
MAKECHAIR(PLANT2,FANCY,MAXLRG)       .50000      5.1798      2.5899
   =L=                                                        =L=
RHS COEFF                                                   140.00

SUM EQUALS                                                  90.213
SHADOW PRICE                                                47.770

##  RESOURCEQ(PLANT2,LRGLATHE)

  VAR                                   Aij         Xj       Aij*Xj
MAKECHAIR(PLANT2,FANCY,NORMAL)       .70000      73.020      51.114
MAKECHAIR(PLANT2,FANCY,MAXLRG)       1.5000      5.1798      7.7697
   =L=                                                        =L=
RHS COEFF                                                   90.000

SUM EQUALS                                                  58.883
SHADOW PRICE                                                38.830
```

Figure 10. Advisory Output

```
 ----#### Executing ADVISORY

 ----### THESE VARIABLES ARE POTENTIALLY UNBOUNDED
             To find the cause of unboundedness
             bound them or the objective function
             variable at a large value. Then solve and
             manually or through GAMSCHK find
             large levels for variables in solution
    SELL(PLANT1,TABLES,FUNCTIONAL)
    SELL(PLANT1,TABLES,FANCY)
    SELL(PLANT1,DINSETS,FUNCTIONAL)
    SELL(PLANT1,DINSETS,FANCY)
    SELL(PLANT2,CHAIRS,FUNCTIONAL)
    SELL(PLANT2,CHAIRS,FANCY)
    SELL(PLANT3,TABLES,FUNCTIONAL)
    SELL(PLANT3,TABLES,FANCY)
    SELL(PLANT3,CHAIRS,FUNCTIONAL)
    SELL(PLANT3,CHAIRS,FANCY)


 ----#### Executing NONOPT

  User comments

    generate output for figure 11


 ----### THESE VARIABLES ARE POTENTIALLY UNBOUNDED
             To find the cause of unboundedness
             bound them or the objective function
             variable at a large value. Then solve and
             manually or through GAMSCHK find
             large levels for variables in solution
    SELL(PLANT1,TABLES,FUNCTIONAL)
    SELL(PLANT1,TABLES,FANCY)
    SELL(PLANT1,DINSETS,FUNCTIONAL)
    SELL(PLANT1,DINSETS,FANCY)
    SELL(PLANT2,CHAIRS,FUNCTIONAL)
    SELL(PLANT2,CHAIRS,FANCY)
    SELL(PLANT3,TABLES,FUNCTIONAL)
    SELL(PLANT3,TABLES,FANCY)
    SELL(PLANT3,CHAIRS,FUNCTIONAL)
    SELL(PLANT3,CHAIRS,FANCY)
```

Figure 11. NONOPT Output - Identify Option

Panel A - ADVISORY Type Output

```
---#### Executing NONOPT

 ----### THESE EQUATIONS ARE POTENTIALLY INFEASIBLE

     RESOURCEQ(PLANT1,LABOR)
     RESOURCEQ(PLANT1,TOP)
     RESOURCEQ(PLANT2,SMLLATHE)
     RESOURCEQ(PLANT2,LRGLATHE)
     RESOURCEQ(PLANT2,CARVER)
     RESOURCEQ(PLANT2,LABOR)
     RESOURCEQ(PLANT3,SMLLATHE)
     RESOURCEQ(PLANT3,LRGLATHE)
     RESOURCEQ(PLANT3,CARVER)
     RESOURCEQ(PLANT3,LABOR)
     RESOURCEQ(PLANT3,TOP)
```

Panel B - Filtered Variables and Equations

```
 ----### THESE VARIABLES ARE POTENTIALLY UNBOUNDED
         Since their levels are so large

   SELL(PLANT2,CHAIRS,FUNCTIONAL)                 level    .1000000E+10
   SELL(PLANT2,CHAIRS,FANCY)                      level    .1000000E+10
   SELL(PLANT3,CHAIRS,FUNCTIONAL)                 level    .1000000E+10
   SELL(PLANT3,CHAIRS,FANCY)                      level    .1000000E+10

 ----### THESE VARIABLE BOUNDS MAY CAUSE INFEASIBLE
         Since their marginals are so large

   MAKECHAIR(PLANT2,FUNCTIONAL,NORMAL)            marg    -.2700000E+10
   MAKECHAIR(PLANT2,FUNCTIONAL,MAXSML)            marg    -.2950000E+10
   MAKECHAIR(PLANT2,FUNCTIONAL,MAXLRG)            marg    -.3000000E+10
   MAKECHAIR(PLANT2,FANCY,NORMAL)                 marg    -.3700000E+10
   MAKECHAIR(PLANT2,FANCY,MAXSML)                 marg    -.3820000E+10
   MAKECHAIR(PLANT2,FANCY,MAXLRG)                 marg    -.3840000E+10
   MAKECHAIR(PLANT3,FUNCTIONAL,NORMAL)            marg    -.2700000E+10
   MAKECHAIR(PLANT3,FUNCTIONAL,MAXSML)            marg    -.2950000E+10
   MAKECHAIR(PLANT3,FUNCTIONAL,MAXLRG)            marg    -.3000000E+10
   MAKECHAIR(PLANT3,FANCY,NORMAL)                 marg    -.3700000E+10
   MAKECHAIR(PLANT3,FANCY,MAXSML)                 marg    -.3820000E+10
   MAKECHAIR(PLANT3,FANCY,MAXLRG)                 marg    -.3840000E+10
   MAKETABLE(PLANT1,FUNCTIONAL)                   marg    -.3999999E+10
   MAKETABLE(PLANT1,FANCY)                        marg    -.5999999E+10
   MAKETABLE(PLANT2,FUNCTIONAL)                   marg    -.3000000E+10
   MAKETABLE(PLANT2,FANCY)                        marg    -.5000000E+10
   MAKETABLE(PLANT3,FUNCTIONAL)                   marg    -.4000000E+10
   MAKETABLE(PLANT3,FANCY)                        marg    -.5999999E+10
   ARTRES(PLANT2,TOP)                             marg    -.1000000E+10

 ----### THESE EQUATIONS ARE POTENTIALLY UNBOUNDED
         Since their levels are so large

   TRNCHAIREQ(PLANT2,FUNCTIONAL)                  level    .1000000E+10
   TRNCHAIREQ(PLANT2,FANCY)                       level    .1000000E+10
   TRNCHAIREQ(PLANT3,FUNCTIONAL)                  level    .1000000E+10
   TRNCHAIREQ(PLANT3,FANCY)                       level    .1000000E+10

 ----### THESE EQUATIONS MAY CAUSE INFEASIBLE
         Since their marginals are so large

   RESOURCEQ(PLANT1,SMLLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT1,LRGLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT1,CARVER)                        marg     .1000000E+10
   RESOURCEQ(PLANT1,LABOR)                         marg     .1000000E+10
   RESOURCEQ(PLANT1,TOP)                           marg     .1000000E+10
   RESOURCEQ(PLANT2,SMLLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT2,LRGLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT2,CARVER)                        marg     .1000000E+10
   RESOURCEQ(PLANT2,LABOR)                         marg     .1000000E+10
   RESOURCEQ(PLANT3,SMLLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT3,LRGLATHE)                      marg     .1000000E+10
   RESOURCEQ(PLANT3,CARVER)                        marg     .1000000E+10
   RESOURCEQ(PLANT3,LABOR)                         marg     .1000000E+10
   RESOURCEQ(PLANT3,TOP)                           marg     .1000000E+10
```

Figure 12. NONOPT Output - Unbounded Model

```
---### LISTING NONOPTIMAL VARIABLES

MAKETABLE (PLANT1, FUNCTIONAL)
   Level       .000000000E+00 Marginal  -80.000000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

MAKETABLE (PLANT1, FANCY)
   Level      .000000000E+00 Marginal  -87.00000000
   Low Bound .000000000E+00 Up Bound .30000000E+31

TRNSPORT (PLANT2, CHAIRS, FUNCTIONAL)
   Level       .000000000E+00 Marginal  -87.000000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

TRNSPORT (PLANT2, CHAIRS, FANCY)
   Level       .000000000E+00 Marginal  -110.000000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

TRNSPORT (PLANT3, TABLES, FUNCTIONAL)
   Level       .000000000E+00 Marginal  -100.000000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

TRNSPORT (PLANT3, TABLES, FANCY)
   Level       .000000000E+00 Marginal  -120.000000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

TRNSPORT (PLANT3, CHAIRS, FUNCTIONAL)
   Level       .000000000E+00 Marginal  -89.000000
   Low Bound  .000000000E+00 Up Bound .30000000E+31

TRNSPORT (PLANT3, CHAIRS, FANCY)
   Level       .000000000E+00 Marginal  -112.00000
   Low Bound  .000000000E+00 Up Bound .300000000E+31

----### LISTING UNBOUNDED EQUATIONS

   TRNCHAIREQ (PLANT2, FANCY)
      Slack  .00000000E+00 Dual 105.00000
      RHS    .00000000E+00
```

Figure 13. NONOPT Output - Infeasible Model

```
 ----#### Executing NONOPT

 ----###  LISTING INFEASIBLE VARIABLES

     MAKETABLE(PLANT2,FUNCTIONAL)
       Level     -41.666667     Marginal  .00000000E+00
       Low Bound  .00000000E+00 Up Bound  .30000000E+31

  ----###   LISTING INFEASIBLE EQUATIONS

     RESOURCEQ(PLANT1,LABOR)
       Slack -175.00000     Dual  .00000000E+00
       RHS   -175.00000

     RESOURCEQ(PLANT1,TOP)
       Slack -50.000000     Dual  .00000000E+00
       RHS   -50.000000

     RESOURCEQ(PLANT2,SMLLATHE)
       Slack -140.00000     Dual  .00000000E+00
       RHS   -140.00000

     RESOURCEQ(PLANT2,LRGLATHE)
       Slack -90.000000     Dual  .00000000E+00
       RHS   -90.000000

     RESOURCEQ(PLANT2,CARVER)
       Slack -120.00000     Dual  .00000000E+00
       RHS   -120.00000

     RESOURCEQ(PLANT3,SMLLATHE)
       Slack -130.00000     Dual  .00000000E+00
       RHS   -130.00000

     RESOURCEQ(PLANT3,LRGLATHE)
       Slack -100.00000     Dual  .00000000E+00
       RHS   -100.00000

     RESOURCEQ(PLANT3,CARVER)
       Slack -110.00000     Dual  .00000000E+00
       RHS   -110.00000

     RESOURCEQ(PLANT3,LABOR)
       Slack -210.00000     Dual  .00000000E+00
       RHS   -210.00000

     RESOURCEQ(PLANT3,TOP)
       Slack -40.000000     Dual  .00000000E+00
       RHS   -40.000000
```

Appendix A: Reserved Names

VARIABLE
VARIABLES
EQUATION
EQUATIONS
INVARIABLE
INVARIABLES
INEQUATION
INEQUATIONS
LISTVARIABLE
LISTVARIABLES
LISTEQUATION
LISTEQUATIONS
POSTOPT
DISPLAYCR
PICTURE
BLOCKPIC
ANALYSIS
MATCHIT
BLOCKLIST
NONOPT
INSOLUTION
NOTINSOLUTION
NONINSOLUTON
VERBOSE
ADVISORY
BINDING
NONBINDING
NOTBINDING
INTERSECT
IDENTIFY
PW=

Appendix B: Sample Problem Used

```
 1
 2
 3
 4
 5
 6  *  SUPPRESS THE PRINTOUT OF SYMBOL LIST
 7  *    AND CROSS REFERENCE MAP
 8
10
11  *  SUPPRESS PRINTOUT OF MODEL ROWS AND COLUMNS
12
13  OPTION LIMROW = 0
14  OPTION LIMCOL =0
15
16  *     SECTION A       SET DEFINITON
17
18   SET  PRODUCT   TABLES CHAIRSSETS    /TABLES, CHAIRS, DINSETS/
19        TYPE      TYPES OF PRODUCT     /FUNCTIONAL ,FANCY/
20        RESOURCE  TYPES OF RESOURCES /SMLLATHE,LRGLATHE,CARVER,LABOR,TOP/
21        METHOD    PRODUCTION METHODS   /NORMAL,MAXSML,MAXLRG/
22        PLANT     DIFFERENT PLANTS     /PLANT1, PLANT2, PLANT3/
23        SUBPRODUCT(PRODUCT)               /TABLES, CHAIRS/;
24
25  *     SECTION B       DATA DEFINITION
26
27   PARAMETER  SETCHAIR(TYPE)  CHAIRS CONTAINED IN EACH SET
28                            / FUNCTIONAL 4, FANCY 6/
29              TABLECOST(TYPE)  TABLECOST  /FUNCTIONAL 80,FANCY 100/;
30
31   TABLE  CHAIRCOST(METHOD,TYPE) CHAIR COST FOR DIFFERENT METHOD
32                                 FUNCTIONAL      FANCY
33              NORMAL                15            25
34              MAXSML                16            25.7
35              MAXLRG                16.5          26.6 ;
36
37   TABLE TB1(RESOURCE,TYPE,METHOD)   USE OF RESOURCES IN CHAIR PRODUCTION
38
39              FUNCTIONAL.NORMAL   FUNCTIONAL.MAXSML   FUNCTIONAL.MAXLRG
40  SMLLATHE          0.8               1.30               0.20
41  LRGLATHE          0.5               0.20               1.30
42  CARVER            0.4               0.40               0.40
43  LABOR             1.0               1.05               1.10
44   +
45
46                 FANCY.NORMAL       FANCY.MAXSML        FANCY.MAXLRG
47  SMLLATHE          1.2               1.7                 0.50
48  LRGLATHE          0.7               0.30                1.50
49  CARVER            1.0               1.00                1.00
50  LABOR             0.8               0.82                0.84;
51
52   TABLE TB2(RESOURCE,TYPE)     USE OF RESOURCES IN TABLE PRODUCTION
53
54                  FUNCTIONAL     FANCY
55     LABOR           3            5
56     TOP             1            1;
57
```

```
 58   TABLE TRANSCOST(SUBPRODUCT, PLANT,TYPE) TRANSPORT COST TO PLANT1
 59
 60               PLANT1.FUNCTIONAL     PLANT2.FUNCTIONAL  PLANT3.FUNCTIONAL
 61      CHAIRS                                  5                7
 62      TABLES                                                   20
 63     +
 64               PLANT1.FANCY     PLANT2.FANCY     PLANT3.FANCY
 65      CHAIRS                          5                7
 66      TABLES                                          20;
 67
 68   TABLE PRICE(PRODUCT,TYPE) PRICE OF CHAIRS
 69                     FUNCTIONAL     FANCY
 70           CHAIRS       82            105
 71           TABLES      200            300
 72           DINSETS     600           1100;
 73
 74   TABLE RESORAVAIL(RESOURCE,PLANT) RESOURCES AVAILABLE
 75                   PLANT1      PLANT2      PLANT3
 76       TOP          50                       40
 77       SMLLATHE                140         130
 78       LRGLATHE                 90         100
 79       CARVER                  120         110
 80       LABOR        175        125         210;
 81
 82   TABLE ACTIVITY(PRODUCT,PLANT) TELLS IF A PLANT SELLS A PRODUCT
 83              PLANT1   PLANT2   PLANT3
 84    TABLES      1                 1
 85    CHAIRS                1        1
 86    DINSETS     1                    ;
 87
 88  *   SECTION     C      MODEL  DEFINITION
 89
 90   POSITIVE VARIABLES
 91      MAKECHAIR(PLANT, TYPE,METHOD)    NUMBER OF CHAIRS MADE
 92      MAKETABLE(PLANT, TYPE)           NUMBER OF TABLES MADE
 93      TRANSPORT(PLANT,SUBPRODUCT,TYPE) NUMBER OF ITEMS TRANSPORTED
 94      SELL(PLANT,PRODUCT,TYPE)         NUMBER OF ITEMS SOLD;
 95   VARIABLES
 96      NETINCOME                        NET REVENUE (PROFIT);
 97   EQUATIONS
 98      OBJT                             OBJECTIVE FUNCTION ( NET REVENUE )
 99      RESOURCEQ(PLANT,RESOURCE)        RESOURCE LIMITS
100      LINKTABLE(TYPE)                  FIRM TABLE LINKAGE CONSTRAINTS
101      LINKCHAIR(TYPE)                  FIRM CHAIR LINKAGE CONSTRAINTS
102      TRNCHAIREQ(PLANT,TYPE)           CHAIRS BALANCE FOR A PLANT
103      TRNTABLEEQ(PLANT,TYPE)           TABLES BALANCE FOR A PLANT;
104
105  OBJT..      NETINCOME =E=
106    SUM((TYPE, PRODUCT, PLANT)$ACTIVITY(PRODUCT,PLANT),
107              PRICE(PRODUCT,TYPE) * SELL(PLANT,PRODUCT,TYPE))
108   -SUM((PLANT,TYPE)$ACTIVITY("TABLES",PLANT),
109              MAKETABLE(PLANT,TYPE)*TABLECOST(TYPE))
110   -SUM((PLANT,TYPE,METHOD)$ACTIVITY("CHAIRS",PLANT),
111              MAKECHAIR(PLANT,TYPE,METHOD)* CHAIRCOST(METHOD,TYPE))
112   -SUM((PLANT,TYPE,SUBPRODUCT)$TRANSCOST(SUBPRODUCT,PLANT,TYPE),
113             TRANSCOST(SUBPRODUCT,PLANT,TYPE)
114           * TRANSPORT(PLANT,SUBPRODUCT,TYPE));
115
```

```
RESOURCEQ(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),
         TRANSPORT(PLANT,"TABLES",TYPE));

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

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

TRNTABLEEQ(PLANT,TYPE)..
  (TRANSPORT(PLANT,"TABLES",TYPE) + SELL(PLANT,"TABLES",TYPE)
  -MAKETABLE(PLANT,TYPE))$TRANSCOST("TABLES",PLANT,TYPE)
              =L= 0 ;
 MODEL FIRM /ALL/;
 MAKECHAIR.up(PLANT, TYPE,METHOD) =5000;
 option solprint = on ;
* SECTION D    SOLVE THE PROBLEM
 option lp=gamschk
 SOLVE FIRM USING LP MAXIMIZING NETINCOME;
```

Appendix C: GAMSCHK One Page Summary

Invoking GAMSCHK **OPTION LP=GAMSCHK**

Keywords allowed in GCK file

Keyword	Allowed SubKEYWORDS	Brief Description
DISPLAYCR	VARIABLE* INVARIABLE* EQUATION* INEQUATION* INTERSECT++	Displays coefficients of selected variables and equations Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations Show coefficients which appear at intersections of selected var/eqn
MATCHIT	VARIABLE* LISTVARIABLE* EQUATION* LISTEQUATION*	List variable and equation names and summarize characteristics Summarizes all variables matching selection statements Lists each variable matching a selection statement Summarizes all equations matching a selection statement Lists each equation matching a selection statement
ANALYSIS		Checks for obvious structural defects
BLOCKLIST		Summarizes characteristics of variable and equation blocks
BLOCKPIC		Generates block level schematics
PICTURE	VARIABLE* INVARIABLE* EQUATION* INEQUATION*	Generates tableau schematics Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations
POSTOPT	VARIABLE* INVARIABLE* EQUATION* INEQUATION* INTERSECT++ NOTINSOLUTION++ INSOLUTION++ BINDING++ NONBINDING++	Reconstructs reduced cost and equation activity Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations Show coefficients which appear at intersections of selected var/eqn Only nonzero vars or those with zero reduced cost Only zero vars will be selected Only eqns with zero slack will be computed Only eqns with nonzero slack will be computed
ADVISORY		List potential infeasible and unbounded items
NONOPT	IDENTIFY VERBOSE	Lists potential or actual nonoptimal items Same as ADVISORY but after solution Does POSTOPT computations on nonoptimals

Other Notes

1) Items marked above with an * are followed by item selection statements.
2) Items marked with ++ modifiy the types of variables, equations and coefficients selected.
3) In item selection an * is a wild card for multiple characters while a . is a wildcard for one character.
4) Spaces and capitalization don't matter in any of the input.
5) Options file controls scaling, solver choice, nonopt filters and maximum allowed selections.
6) Page width is controlled by a PW= keyword but cannot exceed GAMS page width.
7) Lines beginning with a ? or a # are treated as comments.