Misbehaving Mathematical Programs:
Post Optimality Procedures and GAMS Related Software

Bruce A. McCarl

Professor of Agricultural Economics
Texas A&M University
mccarl@tamu.edu
(409)845-1706

A shorter version is Forthcoming in the Journal of Agricultural and Applied Economics

This research was supported by the Texas Agricultural Experiment Station. Thanks to Alex Meeraus and Ramesh Ramen for GAMS support. Thanks to Chi-Chung Chen, Bill Nayda, Claudio Souza and Ruby Ward for comments during software development. Thanks to Harvey Greenberg for comments on the writing.

Misbehaving Mathematical Programs:

Post Optimality Procedures and GAMS Related Software

Mathematical programming is a frequently used tool in agricultural economics analysis. During the past 40 years many papers and books have discussed applications, solution methods, and model formulation techniques. However, few papers discuss model verification and repair.

Some authors have addressed the topic while dealing with other issues. For example, McCarl and Apland (p. 162), in their paper on validation, state that "inconsistent data, bad coefficient placement, incomplete structure and/or an incorrect objective function" can cause improper model results. However, they do not give guidance as to problem detection. In an associated paper, McCarl (p. 161) states, "If the model has failed [validation] discover why....Repair the model and [solve it again]." Pannell (p. 222) argues that "It can be extremely difficult and time consuming to obey ... [McCarl's] simple instructions." However, Pannell while stating that verification has been given "cursory treatment"(p. 20-22) lists the following approaches without providing implementation details(p. 238-40):

a) search for bugs in the matrix;
b) add constraints to force activities to correspond to expected levels;
c) vary right hand sides;
d) drop out matrix components;
e) do a wide ranging sensitivity analysis; and
f) use McCarl and Apland's feasibility test.

The purpose of this paper is to expand upon McCarl and Apland, and Pannell as well as a number of others (Greenberg(1993,1994); Chinneck; Andersen and Andersen) providing systematic, computerized approaches for the detection of model flaws.

The procedures herein are available in computerized form linked to GAMS. GAMS (Brooke, Kendrick and Meeraus) is the most widely used method for implementing mathematical programs by agricultural economists. The package is called GAMSCHK and is available through the web page agrinet.tamu.edu/mccarl. The GAMSCHK software has both pre- and post-solution processors to aid in discovering model problems. This manuscript only covers post-solution checking.[1]

Background

Many modelers finish their model and submit it to a solver only to find that the model is infeasible, unbounded, or worse yet, optimal with an unrealistic solution. Generally, such problems are caused by the litany of difficulties mentioned in McCarl and Apland, or Pannell (p. 228) including models with:

a) incorrect coefficients due to typing, sign, units of measurement, calculation flaws, omissions, or improper placement;

b) improper constraints in terms of inequality forms, coefficient/constraint omissions, unneeded redundancies, flawed coefficient calculations, or inclusion of binding but irrelevant constraints;

c) an improper set of variables with irrelevant variables included, relevant variables excluded or improperly entered variables in terms of coefficient signs, placements or magnitudes; and

d) a structure that causes solvers to fail.

This paper discusses methods for discovery of the above cases excepting solver failure. The presentation will concentrate on using knowledge of the problem, mathematical programming theory, and solution information to systematically discover difficulties.

Problematic Model Substructures – the Cause of Difficulties

Flawed model structural characteristics cause flawed solutions. Greenberg, in a series of papers (1993,1994,1996), defined the concept of forcing substructures, wherein a subset of constraints and associated variables forces a set of variables to equal particular values. A related concept was developed by Chinneck who defined an irreducible infeasible set as a set of constraints that when anyone is removed changes a model from being infeasible to being feasible. Herein, a broader term "problematic substructure" (PS), is used to refer to a portion of a formulation where the associated variables, constraints, and coefficients cause an improper solution.

Linear Programming Theory

It is useful in this paper to employ several linear programming theoretical results. These are presented following the notation in Hadley or Bazarra, Jarvis and Sherali. Consider a linear programming problem with slack variables (S_i) added:

$$
\begin{array}{llll}
(1) & \text{Maximize} & \sum_j c_j X_j + \sum_i 0 \cdot S_i & \\
 & \text{subject to} & \sum_j a_{ij} X_j + S_i = b_i & \text{for all } i \\
 & & X_j, \quad S_i \geq 0 & \text{for all } i,j
\end{array}
$$

At optimality given identification of a basis matrix (B) and the associated basic elements from the objective function (C_B), an expression for the optimal shadow prices on the i^{th} equation is:

$$(2) \qquad u_i = (C_B B^{-1})_I$$

while an expression for the reduced cost of the j^{th} variable is:

(3) $$\sum_i u_i a_{ij} - c_j \geq 0$$

Note that for basic variables equation (3) equals zero and that equation (2) arises from the solution of that system. Also note that any feasible solution must satisfy the original set of constraints and thus:

(4) $$\sum_j a_{ij} X_j - b_i = -S_i \text{ for all } i$$

while the optimal values of the basic variables are given by:

(5) $$X_B = B^{-1}b.$$

Finally, following the development in Chapter 3 of McCarl and Spreen and differentiating (5) we get an expression for the marginal effect of the right hand side changes.

(6) $$\frac{\partial X_B}{\partial b} = B^{-1}$$

Equations 2-6 are used in explaining the procedures below.

Adopting an Example

The discussion of how to fix misbehaving models is best facilitated by having one. The example in Figure 1 will provide the base for later illustrations. This example has two farms, each of which can feed cattle and grow crops. The feeding requirements, livestock costs, and final sale weights, costs, crop yields, and sale prices differ by farm. Crops can be transported between farms. Land can be rented on each farm. In the solution the variable labeled PROFIT is maximized. A GAMS formulation appears in Appendix A.

Unbounded Models

The solution report on an unbounded linear program is often lacking information. More information can be gained by artificially bounding the problem, then using solution information to find the PS. In particular, following the suggestion in both McCarl and Apland; and Brooke, Kendrick and Meeraus, a large upper bound is assigned to all variables that have desirable objective function values which are not already bounded. Given a problem with the structure in (1), then for all variables with $c_j > 0$ imposition of large positive upper bounds will render the problem bounded. The resultant needed bounds, where M is a very large number are:

$$(7) \qquad X_j \leq M \qquad \text{for all } j \text{ where } c_j > 0$$

The model is then solved and the solution examined for variables which equal the large bound. However, those variables only compose part of the PS. In particular, suppose a large bound is active in the k^{th} row. Also suppose that the right hand side (the M value in equation 7) equals the right hand side which would generate the expected solution variable values (X_B^e) plus 10^{10}. Then via (5) the optimal value of the basic variables is

$$(8) \qquad X_B = B^{-1}b = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1k} & \cdots & \gamma_{1m} \\ \vdots & & & & \\ \gamma_{k1} & \cdots & \gamma_{kk} & \cdots & \gamma_{km} \\ \vdots & & & & \\ \gamma_{m1} & \cdots & \gamma_{mk} & \cdots & \gamma_{mm} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_k + 10^{10} \\ \vdots \\ b_m \end{bmatrix} = X_B^e + 10^{10}\gamma_k$$

where γ_{ik} is the ik^{th} element of B^{-1}, and γ_k is the vector from the k^{th} column of B^{-1}. Thus, the solution is composed of numbers of the magnitude expected plus $(_k$ coefficients times the large term in the bound. Via equation 6 the $(_{ik}$ give the expected change in the i^{th} basic variable when the k^{th} right hand side is changed by one unit. Consequently, the $(_k$ vector indicates how the alteration of the large bound affects each basic variable. The variables with large solution values are those which are associated with the original unboundedness and form the PS.

This is best illustrated by example. The Figure 1 example is made unbounded by dropping the constraints on maximum rented land and enter the cattle price on Farm 1 in cents per pound, making a units error(see lines 87-88 in the appendix). Under these circumstances the model looks like that in Figure 2. That model is unbounded, so bounds are included on the revenue producing cattle production and crop sale variables. (See lines 89-90 of the appendix)

The resultant solution is appended around the edges of Figure 2. The feed cattle alternative on Farm 1 is at its large upper bound. But also notice: a) large crop acreages on Farm 1, and b) large land rental on Farm 1.

The PS is composed of the variables with large values and involves cattle feeding, crop growing, and land rental all on Farm 1. One then would examine only those variables and any rows where more than one of them appears. A modeler would then find the unboundedness cause immediately apparent. Namely the PS is in the huge objective function coefficients for fed cattle which arise either because of the cattle sale price, improper units for cattle sale weight (if the sale weight were in hundred weight the problem would disappear), and/or the lack of upper bounds on renting land.

This is indicative of the general approach to finding difficulties in unbounded models of the type in (1)[2]. The approach is as follows:

Step 1 Given an unbounded model add larger upper lower bounds to all profitable variables (in a maximization problem this would be all variables with a positive objective function value).

Step 2 Solve the model.

Step 3 See if any of the large upper bounds that were added are binding. If not terminate the unboundedness finding procedure. If so, go to step 4.

Step 4 Select variables and slacks whose solution values in absolute value are greater than or equal to a tolerance which is set to be substantially greater than the expected order of magnitude of the solution variables. Examine that set of variables and equations as well as any interrelating constraints to find the cause of the unboundedness.

Step 5 Fix the model and reexecute the procedure.

Several notes are in order about this procedure. First, the bounds must be large enough so that they will cause the solution to have unrealistically large values for some variables. This may take experimentation either increasing the bound value or rescaling other model components. Second, a variant of this procedure can be utilized in GAMS where one bounds only the objective function variable (the variable named PROFIT in the example). However, such a procedure will only find one case of the unboundedness at a time. When multiple bounds are added multiple unbounded cases can be found in one pass. Third, GAMSCHK facilitates this procedure, and when run in "NONOPT" mode it automatically: a) identifies all variables which need bounds when an unbounded model is present, and b) displays the names of variables or equations where the optimal solution values exceed a tolerance in an optimal solution. In turn, the user can use GAMSCHK to extract the potential PS equations and variables. Fourth, as illustrated above, the problematic structure can involve multiple interrelated model elements. Experience shows that in problems with thousands of variables and equations one may find 5-6 variables and constraints in the PS. Reduction to a small PS makes finding the PS cause relatively simple. Fifth, the above example shows the potential complexity of PSs which cause unboundedness. The unboundedness could have been caused by improper cattle sale weights, cattle prices, or land rental limits. Unboundedness does not necessarily occur strictly because of the variable which the solvers report as unbounded or which hit the large upper bound, but rather may occur because of interrelationships with other variables. Sixth, the burden involved in adding bounds when using GAMS is not high because one can include statements such as in lines 89-90 which add bounds to all variables in an variable block.

Infeasible Models

Infeasibility causing PS can be found using the traditional Big "M" artificial variable approach with the added step of shadow price examination to find the full PS. In using Big M the model is augmented with artificial variables added to any constraint which is not satisfied when the original problem decision variables are equal to zero. This includes all constraints which require a sum of the decision variables to be: a) greater than or equal to a positive right hand side; b) less than or equal to a negative right hand side; and c) equal to a non-zero quantity. The artificial variables also have an associated large penalty cost in the objective function, which makes them highly undesirable to have in the optimal solution.

When the original problem is infeasible, then one or more of these added artificials will be nonzero. The presence of nonzero artificials in the basis distorts the shadow price and reduced cost solution. Shadow prices in a linear programming model are given by $C_B B^{-1}$. When some elements in C_B are large penalties associated with artificials then some shadow prices and reduced costs will also be affected receiving large values[3]. The constraints and nonnegativity conditions/bounds associated with these large values are the infeasibility causing PS.

Again this is best illustrated through example. Models are infeasible because constraints are mutually inconsistent. The example is rendered infeasible by increasing the land requirement for cattle up to 10 (as in line 93 of the appendix). Artificial variables are also defined for the cattle minimum sales requirements[4].

The solution shows large shadow prices for Farm 1 land, maximum rented land, and minimum cattle sales, as well as large crop growing reduced costs. The PS involves this set of constraints and nonnegativity conditions. Namely, the cattle sales constraint on Farm 1 cannot be satisfied with 10 units of land used per head given the availability of owned and rented land along with the nonnegativitiy restrictions on the crop variables. Repair of this problem could entail: a) reduction of the cattle land use requirement, b) increase in the land available, and c) respecification of the model to include pasture land and shift in the cattle land requirement to that resource. This is again illustrative of the general nature of PSs. In particular, the cause is not always the constraint in which the artificial variable is active (i.e., not the minimum cattle sales from Farm 1), rather the cause usually involves the interaction of several constraints.

A general approach for finding infeasibility causing PS:

Step 1 Take an infeasible model and enter artificial variables in all equations that are not feasible when the variables are set to zero. Add Big "Ms" for the artificial variables in the objective function. Artificial variables could also be needed for positive lower bounds and negative upper bounds.

Step 2 Solve the model.

Step 3 See if any of the artificial variables are in the basis, if not, terminate. If so, go to step 4.

Step 4 Find all equations as well as variable upper and lower bounds with shadow prices that are large in absolute value. These are the PS. Examine them and the variables there in to find data errors.

Step 5 Fix any errors that are found, and repeat the process if needed.

Several comments are in order about the above procedure. First, the Big “M” penalties must be large enough to distort the shadow prices and this may take several iterations either increasing the M value or rescaling other model components. Second, if infeasibility causing constraints are redundant, this procedure may not discover all of the PS in one pass. Third, in the above example, a part of the PS involved the lower bounds on the crop growing variables. The model would like to drive those variables negative which would make them a source of labor. PSs may include upper and lower (including nonnegativity) bounds indicating that reduction of the lower bound or increase of the upper bound would help alleviate the infeasibility. One occasionally has to take the discovery of these items with a “grain of salt”. Fourth, the above procedures can be used easily, but are not needed if one is using a solver that has Chinneck’s IIS capability. In particular, the GAMS CPLEX version has IIS capabilities. Use of Greenberg’s ANALYZE (1993) also substitutes for the above procedures. Fifth, GAMSCHK will both identify constraints where artificial variables are needed as well as find equations and variables with shadow prices and reduced costs greater in absolute value then a tolerance. Sixth, ultimately, the resolution of any infeasibility will require manual examination of the constraints to find the PS cause. Use of the above procedure allows one to restrict attention to a small model subset. Seventh, addition of the artificials is more complicated than the bounds above but can still be done in a relatively few statements using the algebraic capabilities of GAMS.

The Big M method is covered in many texts and papers; however, those authors do not discuss how to find the infeasibility causing constraints. Greenberg (1994) suggests the use of Phase I shadow prices from the linear programming solver to diagnose infeasibility and it turns out that the artificial variable distortion will occur in the rows with Phase 1 shadow prices. Finally, note that this approach is the dual to the large bounds approach, namely placing

artificials in the primal is equivalent to putting large bounds on the dual or the converse.

Models with Unrealistic Solutions

Unfortunately, unbounded and infeasible cases are typically the easy cases of model diagnosis. One receives obvious notification from the solver that there is something wrong and can find the PS after adding artificials or bounds. More difficult cases arise when one gets an "optimal solution", but discovers that the solution substance is unrealistic. This often necessitates a rather involved model investigation, and always requires expectations about the appropriate levels of variables and shadow prices.

An unrealistic optimal solution can emerge through unrealistic allocation or valuation information. A model that contains unrealistic allocation information has faulty levels for some variables and/or slacks. A model with unrealistic valuation information contains faulty reduced costs and/or shadow prices.

Modelers can find unrealistic solution causing PS through examination of either: a) valuation information using what is called budgeting herein or b) allocation information using a process called row summing herein.

Finding causes of Unrealistic Solutions through Budgeting

If the solution contains improper valuation information, then the shadow prices are wrong as may be some reduced costs (as in equation 3). Shadow prices are determined by the solution of the reduced cost equations (3) for the basic variables. Thus, the causes of faulty valuation information can be discovered by examining the reduced cost calculations. For the j^{th} variable this is done by creating an expanded version of the reduced cost calculations as follows:

Item	**a_{ij}**	**Shadow Price**	**Product**
Objective function	c_j	-1	$-c_j$
Name for equation 1	a_{1j}	u_1	$u_1\ a_{1j}$
Name for equation 2	a_{2j}	u_2	$u_2\ a_{2j}$
	!	!	!
Name for equation m	a_{mj}	u_m	$u_m\ a_{mj}$
Reduced cost	--	--	$E_i u_i a_{ij}-c_j$

This table has a row for each equation in which the j^{th} variable has a non-zero a_{ij}, and includes the a_{ij}, the shadow price for the equation(u_i), and their product. In turn, the products are added, yielding a reproduction of the reduced cost. For example when budgeting, say, corn production, you would record each resource used in corn production, the amount a unit of corn production uses, and the resource prices. Then multiply usage by price yielding total cost for each resource, and add up to get reduced cost.

A model with an unrealistic optimal solution is illustrated by modifying the corn yield for farm 2 so it is pounds rather than bushels, and zeroing the minimum cattle sale requirement (Figure 4 and appendix lines 95-96). Model solution yields the information that appears around the edges of Figure 4. A symptom of the model problems is found in the \$8055 reduced cost of producing Farm 2 cattle. Normal returns would likely fall in the range from \$150 to \$200 per head. The question then is why does it cost so much to grow the cattle? Table 1 panel a contains a budget for that variable.

This cattle variable produces objective function income (c_j) of \$153.2 per unit, uses 38.48 units of corn, 0.74 units of hay, 0.5 units of land, and produces 1 unit of cattle sales. The shadow prices and product columns shows the net revenue of $153.2, is counterbalanced by use of 38.48 units corn worth $2.29/bushel amounting to $88.04 worth of corn, hay worth $39.62, and 0.5 acres of land worth $16,160/acre or $8,080 in total. Summing the reduced cost of raising cattle is $8,054. The land value is problematic, since land contributes most of the $8,054 opportunity cost, and is valued at $16,160. Thus, the land shadow price merits further examination.

Shadow price values are derived from the parameters of the basic variables. Land is so valuable, because of a basic variable that uses land. In this model, the basic variable using farm 2 land is corn production, which is budgeted in panel b, Table 1.

Corn exhibits a $240 cost, a yield, and use of land. The $16,400 opportunity cost of land is shown to be mainly counter balanced by an unrealistic 7168 bushel yield which is valued at $2.29. Thus, the land distortion and the original reduced cost symptom occurs because of the excessively high corn yield. A modeler would then correct that problem, solve the model again and repeat the procedure if other solution irregularities were found.

The general budgeting approach entails the following:

Step 1 Examine the shadow prices and the reduced cost solution to see if any elements are unrealistic.

Step 2 If an unrealistic reduced cost has been found then budget that variable and examine to see if there is a data error or an excessively high shadow price. If a data error is found in the model, correct that problem, resolve the model, and go to Step 1. Otherwise go on to step 3.

Step 3 Given an unrealistic shadow price has been found, budget the basic variables that have non-zero coefficients in the associated equation. Examine those budgets to find either additional unrealistic shadow prices or a data error which is causing the unrealistic shadow price.

Step 4 If another unrealistically high shadow price is found, then go to step 3 and budget another basic variables which uses that resource and iterate through until a data error is found.

Step 5 When a data error is found correct the model, resolve and go to Step 1.

Part of this procedure may involve making sure the appropriate constraints are binding for a variable. In particular, if a constraint that is felt to be binding is not binding or an a_{ij} is missing, one may need to either: investigate the constraint through row summing as discussed below, add a missing constraint, or add missing coefficients.

The above procedure is meritorious of several comments. First, budgeting always requires problem knowledge as well as expectations about the proper values of shadow prices and coefficients. One must understand the model to debug it. Second, one identifies the PS by finding unrealistic shadow prices and then tracing through the model to find coefficient errors, missing coefficients, constraints that should be binding, and other errors. The overall PS is each of the variables budgeted coupled with each equation associated with an improper shadow price. Third, the budgeting procedure has been implemented in the GAMSCHK software. Using that software, one can request that selected variables be budgeted or that all variables that fall into particular equations can be budgeted. Attention can also be restricted to binding equations and basic variables. Fourth, note the analogy between this procedure and traditional financial budgeting. When doing financial budgeting, one ordinarily takes per unit resource usages, multiplies them by prices, then accumulates to come up with a bottom line. The procedure here is exactly analogous, with the prices used being the shadow prices derived by the solver.

Finding Causes of Unrealistic Solutions through Row Summing

The dual approach to budgeting is to look at the primal allocation. When looking at primal allocation one searches for unrealistically high or low variable solution values. Variable values arise through the solution of binding constraints as in equation 4. Thus, to discover why variables have unrealistic values, one looks at the constraints. Row summing involves systematic reconstruction of constraint activity. The basic table for row summing the i^{th} equation is:

Variable Names	**Coefficients**	**Solution Levels**	**Product**
$\mathbf{X_1}$	a_{i1}	X_1^*	$a_{i1}X_1^*$
$\mathbf{X_2}$	a_{i2}	X_2^*	$a_{i2}X_2^*$
!	!	!	!
$\mathbf{X_n}$	a_{in}	X_n*	$a_{in}X_n$*
Sum	--	--	$\Sigma_j a_{ij}X_j$*
RHS	–	—	b_i
Slack	--	--	b_i-Σ_j $a_{ij}X_j$*

The table has a row for each variable that falls into the equation, the a_{ij}, the optimal variable value(X_j^*) and the product ($a_{ij}X_j^*$). In turn, the products are summed, the right hand side recorded and the slack computed. The symptom that causes the use of row summing is an unrealistically high value of a decision variable and/or slack.

To illustrate row summing the example in Figure 4 is reused. Net profits of $12,867,960 are excessive, thus the objective function is row summed (Table 2, panel A).

The objective function row sum shows the profits comes largely from the $13,747,853 farm 1 corn sale. A row sum on the Farm 1 corn balance, appears in Table 2, panel B. That shows the 5,728,272 bushels sold on Farm 1 arises largely from corn transported from Farm 2. Table 2, panel C shows a row sum for the corn balance on Farm 2, and reveals that so much corn can be shipped because of the excessive yield of corn on Farm 2. Thus, the underlying PS involves the sales and the two balance equations.

In general, row summing is employed via the following steps.

Step 1	Find a variable or a slack with an unreasonable solution level, if there are none stop.
Step 2	Choose an equation that is likely to reveal some information where that variable has coefficients.
Step 3	Row sum that equation to discover other variables with unrealistically high solution values and/or coefficient errors. If errors are discovered fix the problem, resolve the model and go to Step 1. Otherwise, go to Step 4.
Step 4	If other variables with unreasonable solution values are discovered, select equations for row summing that contain those variables and go back to Step 2.

The row summing procedure allows systematic exploration of the allocation information to see how values of some variables are balanced off against values of other variables. It also can be used to find incidence of excessive resource use in the model.

Several comments can be made on this procedure. First, one can use budgeting and row summing independently or collectively to discover PS. Second, GAMSCHK will automatically construct row sums on any named equation or any equations wherein a named variable has coefficients. GAMSCHK can be limited to binding equations and basic variables. Third, one must have expectations about appropriate solution values to identify problems. Fourth, row summing may be coupled with expectations about proper structure to find missing coefficients, and/or variables as well as mis-specified coefficients.

Concluding Comments

The procedures presented herein give theoretically based ways to find problematic substructures that result in unboundedness, infeasibility, or unrealistic optimal solutions in mathematical programming models. The unboundedness and infeasibility procedures rely on model augmentation, coupled with analysis of the shadow price or allocation information. The unrealistic optimal solution procedures provide systematic ways of examining models to identify the source of problems. They may also be used in either the unbounded and infeasible cases.

The procedures may be utilized on nonlinear, mixed integer, or linear programming models. In the case of nonlinear models one needs to solve the model and then employ the procedures as opposed to employing the procedures on the initial formulation. This is recommended since codes such as GAMS employ a Jacobian based representation using a local Taylor series expansion around the current point. Once the model is solved, the Taylor series expansion is based on the current solution and is more accurate than it is before solution. GAMSCHK marks nonlinear coefficients to inform users that local values are present.

In mixed integer cases, the procedures can be used in a straight forward manner. However, one must realize that the shadow prices could be distorted because of the noncontinuous nature of the feasible solution space and the solution algorithm wherein often constraints are artificially imposed to generate an integer solution.

The procedures discussed above are all implemented in the GAMSCHK software that is distributed freely through the web-page referenced in the bibliography (McCarl). The procedures discussed above involve post optimality evaluations of potential structural problems. GAMSCHK also contains pre optimality evaluation procedures.

Bibliography

Andersen, E.D. and K. D. Andersen, "Presolving in Linear Programming," *Mathematical Programming*, 71 (1995): 221-245.

Bazarra, M.S., J. J. Jarvis , and H. D. Sherali, *Linear Programming and Network Flows*. John Wiley, New York, 1990.

Brooke, A., D. Kendrick, and A. Meeraus. *GAMS: A User's Guide*. Boyd and Fraser Publishers, Version 2.25, 1993.

Chinneck, J. W., "Feasibility and Viability,"Chapter 14 in *Recent Advances in Sensitivity Analysis and Parametric Programming*, Edited by T. Gal and H. J. Greenberg, Kluwer Academic Publishers 1997.

Greenberg, H. J. "A Bibliography for the Development of an Intelligent Mathematical Programming System", *Interactive Transactions in Operations Research and Management Sciences*, 1:1(1996), http://www-math.cudenver.edu/~hgreenbe/impsbib/impsbib.html.

Greenberg, H. J. *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for Analyze* , Kluwer Publishers Boston, MA, 1993.

Greenberg, H. J., "How to Analyze the Results of Linear Programs-Part 4: Forcing Substructures," *Interfaces*, 24:1(Jan.-Feb. 1994):121-130.

Hadley, G., *Linear Programming*, Addison Wesley, Reading, 1962.

McCarl, B.A. *GAMSCHK USER DOCUMENTATION: A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS*. Web Page URL http://agrinet.tamu.edu/mccarl, Texas A&M University, 1997.

McCarl, B.A. "Model Validation: An Overview with Some Emphasis on Risk Models", *Review of Marketing and Agricultural Economics*,5 2(3) p. 153-173,1984.

McCarl, B. A. and Apland J.D. , "Validation of Linear Programming Models", Southern Journal of Agricultural Economics, 18(December 1986), p. 155-164.

McCarl, B.A. and T. H. Spreen. Applied Mathematical Programming using Algebraic Systems, Draft Textbook, Texas A&M University. Distributed through web page http://agrinet.tamu.edu/mccarl, 1997.

Pannell, D.J., Introduction to Practical Linear Programming, John Wiley &Sons, Inc. New York: 1997.

Figure 1. **Base Example Model**

			Profit	Feed Cattle		Move Crops				Grow Crops				Sell Crops				Rent Land		RHS	
				Farm 1	Farm 2	Farm 1 to Farm 2		Farm 2 to Farm 1		Farm 1		Farm 2		Farm 1		Farm 2					
						Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Farm 1	Farm 2		
Profit Accounting			1	-185	-153	.11	4	.11	4	250	220	240	195	-2.4	-55	-2.05	-50	100	100	= 0	
Crop on Hand	Farm 1	Corn		39		1		-1		-130				1						#	0
		Hay		0.75			1		-1		-5.5				1					#	0
	Farm 2	Corn			38.5	-1		1				-128				1				#	0
		Hay			0.74		-1		1				-4.8				1			#	0
Land	Farm 1			0.5						1	1							-1		#	100
	Farm 2				0.5							1	1						-1	#	100
Min. Cattle Sold	Farm 1			1																$	50
	Farm 2				1															$	50
Max Rented Land	Farm 1																	1		#	200
	Farm 2																		1	#	700

Figure 2. **Unbounded Example**

			Profit	Feed Cattle		Move Crops				Grow Crops				Sell Crops				Rent Land		RHS		Slack	Shadow Price
				Farm 1	Farm 2	Farm 1 to Farm 2		Farm 2 to Farm 1		Farm 1		Farm 2		Farm 1		Farm 2							
						Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Farm 1	Farm 2				
Profit Accounting			1	-76415	-153	.11	4	.11	4	250	220	240	195	-2.4	-55	-2.05	-50	100	100	=	0	0	1
Crop on Hand	Farm 1	Corn		39		1		-1		-130				1						#	0	0	2.69
		Hay		0.75			1		-1		-5.5				1					#	0	0	58.18
	Farm 2	Corn			38.5	-1		1				-128				1				#	0	0	2.58
		Hay			0.74		-1		1				-4.8				1			#	0	0	59.43
Land	Farm 1			0.5						1	1							-1		#	100	0	100
	Farm 2				0.5							1	1						-1	#	100	0	90.28
Min. Cattle Sold	Farm 1			1																$	50	1e+6	0
	Farm 2				1															$	50	0	-35.21
~~Max Rented Land~~	~~Farm 1~~																	~~1~~		~~#~~	~~200~~		
																			~~1~~	~~#~~	~~700~~		
Upper Bounds			--	1e+6	1e+6	--	--	--	--	--	--	--	--	1e+6	1e+6	1e+6	1e+6	--	--				
Optimal Level			7e+10	1e+6	50	0	0	6689	0	3e+5	1e+5	67	8	0	0	0	0	9e+5	0				
Reduced Cost			0	0	0	-.22	-2.75	0	-5.25	0	0	0	0	1e+6	1e+6	1e+6	1e+6	0	-9.72				

Figure 3. **Infeasible Example**

			Profit	Feed Cattle		Move Crops				Grow Crops				Sell Crops				Rent Land		Artificial Cattle		RHS		Slack	Shadow Price
				Farm 1	Farm 2	Farm 1 to Farm 2		Farm 2 to Farm 1		Farm 1		Farm 2		Farm 1		Farm 2									
						Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Farm 1	Farm 2	Farm 1	Farm 2				
Profit Accounting			1	-185	-153	.11	4	.11	4	250	220	240	195	-2.4	-55	-2.05	-50	100	100	1e+6	1e+6	=	0	0	1
Crop on Hand	Farm 1	Corn		39		1		-1		-130				1								#	0	0	2.77
		Hay		0.75			1		-1		-5.5				1							#	0	0	65.46
	Farm 2	Corn			38.5	-1		1				-128				1						#	0	0	2.66
		Hay			0.74		-1		1				-4.8				1					#	0	0	61.46
Land	Farm 1			10						1	1							-1		-1		#	100	0	1e+5
	Farm 2				10							1	1						-1		-1	#	100	0	100
Min. Cattle Sold	Farm 1			1																		$	50	1e+6	-1e+6
	Farm 2				1																	$	50	0	-944.49
Max Rented Land	Farm 1																	1		1		#	200	0	99903
	Farm 2																		1		1	#	700	476	0
Optimal Level			-2e+7	30	50	0	0	1170	22.5	0	0	24	12	0	0	0	0	200	437	20	0				
Reduced Cost			0	0	0	-.22	-8	0	0	-1e+5	-1e+5	0	0	-.37	-10.5	-.61	-11.5	0	0	0	-1e+6				

Figure 4. **Unrealistic Optimal Example**

			Profit	Feed Cattle		Move Crops				Grow Crops				Sell Crops				Rent Land		RHS		Slack	Shadow Price
				Farm 1	Farm 2	Farm 1 to Farm 2		Farm 2 to Farm 1		Farm 1		Farm 2		Farm 1		Farm 2							
						Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Farm 1	Farm 2				
Profit Accounting			1	-185	-153	.11	4	.11	4	250	220	240	195	-2.4	-55	-2.05	-50	100	100	=	0	0	1
Crop on Hand	Farm 1	Corn		39		1		-1		-130				1						#	0	0	2.40
		Hay		0.75			1		-1		-5.5				1					#	0	0	55
	Farm 2	Corn			38.5	-1		1				-7168				1				#	0	0	2.29
		Hay			0.74		-1		1				-4.8				1			#	0	0	51
Land	Farm 1			0.5						1	1							-1		#	100	0	96.49
	Farm 2				0.5							1	1						-1	#	100	0	16160
Min. Cattle Sold	Farm 1			1																$	0	157	0
	Farm 2				1															$	0	0	0
Max Rented Land	Farm 1																	1		#	200	200	0
	Farm 2																		1	#	700	0	16060

Optimal Level	1.3e+7	157	0	0	0	5.7e+6	0	0	21	800	0	5.7e+6	0	0	0	0	700
Reduced Cost	0	0	-8055	-.22	-8	0	0	-34	0	0	-16100	0	-2.54	-.24	-3.54	-3.51	0

Table 1. Budgeting for Unrealistic Optimal Case

Panel a FEEDCATTLE(FARM2) Budget

Equation	A_{ij}	U_i	$A_{ij}*U_i$
PROFITACCT	-153.20	1.00	-153.20
CROPONHAND(FARM2,CORN)	38.48	2.29	88.04
CROPONHAND(FARM2,HAY)	0.74	53.54	39.62
LAND(FARM2)	0.50	16160.40	8080.20
MINCATTLE(FARM2)	1.00	0.00	0.00
TRUE REDUCED COST			8054.66

Panel b GROWCROPS(FARM2,CORN) Budget

Equation	A_{ij}	U_i	$A_{ij}*U_i$
PROFITACCT	240.00	1.00	240.00
CROPONHAND(FARM2,CORN)	-7162.00	2.29	-16400.00
LAND(FARM1)	1.00	96.49	16160.00
TRUE REDUCED COST			0.00

Table 2. Row Summing for Unrealistic Optimal Case

Panel a Objective Function -- PROFITACCT Row Sum

Variable	A_{ij}	X_j	$A_{ij}*X_j$
PROFIT	1.00	12867960	12867960
FEEDCATTLE(FARM1)	-185.00	157.14	-29071
FEEDCATTLE(FARM2)	-153.20	0.00	0
MOVECROPS(FARM1,FARM2,CORN)	0.11	0.00	0
MOVECROPS(FARM1,FARM2,HAY)	4.00	0.00	0
MOVECROPS(FARM2,FARM1,CORN)	0.11	5734400	642250
MOVECROPS(FARM2,FARM1,HAY)	4.00	0.00	0
GROWCROPS(FARM1,CORN)	250.00	0.00	0
GROWCROPS(FARM1,HAY)	220.00	21.43	4714
GROWCROPS(FARM2,CORN)	240.00	800.00	192000
GROWCROPS(FARM2,HAY)	195.00	0.00	0
SELLCROPS(FARM1,CORN)	-2.40	5728272	-13747853
SELLCROPS(FARM1,HAY)	-55.00	0.00	0
SELLCROPS(FARM2,CORN)	-2.05	0.00	0
SELLCROPS(FARM2,HAY)	-50.00	0.00	0
LANDRENT(FARM1)	100.00	0.00	0
LANDRENT(FARM2)	100.00	700.00	70000
=E=			=E=
RHS COEFF			0

Panel b CROPONHAND(FARM1,CORN) Row Sum

Variable	A_{ij}	X_j	$A_{ij}*X_j$
FEEDCATTLE(FARM1)	39.00	157.14	6128
MOVECROPS(FARM1,FARM2,CORN)	1.00	0.00	0
MOVECROPS(FARM2,FARM1,CORN)	-1.00	5734400	5734400
GROWCROPS(FARM1,CORN)	-130.00	0.00	0
SELLCROPS(FARM1,CORN)	1.00	5728272	5728272
=L=			
=L=			
RHS COEFF			0

Panel c CROPONHAND(FARM2,CORN) Row Sum

Variable	A_{ij}	X_j	$A_{ij}*X_j$
FEEDCATTLE(FARM2)	38.48	0.00	0
MOVECROPS(FARM2,FARM1,CORN)	-1.00	5734400	5734400
MOVECROPS(FARM1,FARM2,CORN)	1.00	0.00	0
GROWCROPS(FARM2,CORN)	-7168.00	800.00	5734400
SELLCROPS(FARM2,CORN)	1.00	0.00	0

=L= =L=
RHS COEFF 0


```
option solslack=1;
option limrow=0;
option limcol=0;
Sets
  Farm                  farm names              /farm1,farm2/
  allitems              union of sets where item names reused
                                    /othfeedcst,feedbeef,corn,hay/
  enterprise(allitems) all production enterp   /feedbeef,corn,hay/
  crop(enterprise)      crops                   /corn,hay/
  alias(place,farm);
parameters
 landavail(farm)      land endowment      /farm1   100   ,farm2  100 /
 cashrent(farm)       land rental rate    /farm1   100   ,farm2  100 /
 rentavail(farm)      land avail to rent  /farm1   200   ,farm2  700 /
 cattlepric(farm)     cattle sale price   /farm1  0.70   ,farm2 0.68 /
 cattlecost(farm)     cost cattle rearing /farm1   570   ,farm2  580 /
 cattlewgt(farm)      cattle sale weight  /farm1  1100   ,farm2 1100 /
 cattleday(farm)      days on feed        /farm1   150   ,farm2  148 /
 contract(farm)       minimum cattle fed  /farm1    50   ,farm2   50 /
table cropprice(farm,crop) crop sales price
               corn     hay
     farm1      2.40    55
     farm2      2.05    50
table cropcost(farm,crop) crop production cost per acre
               corn     hay
     farm1     250      220
     farm2     240      195
table cropyield(farm,crop) crop yields in units per acre
               corn     hay
     farm1      130      5.5
     farm2      128      4.8
parameter dietdat(allitems) diets per head per day
     /othfeedcst 0.10,  corn 0.26, hay 0.005/
parameter weight(crop) weight of crops
      /corn 56,hay 2000/
scalar transport transport costs per lb  /0.002/
scalar cattleland cattle land use         /0.5/
variables
  profit                            total firm profits
positive variables
  feedcattle(farm)                  number of cattle fed
  cattlediet(farm,crop)             amount of each crop fed to cattle
  movecrops(farm,place,crop)        transport of crops between places
  growcrops(farm,crop)              raising crops at a farm
  sellcrops(farm,crop)              sell crops at a farm
  landrent(farm)                    land rental
  artcattle(farm)                   cattle artificial variable
equations
  profitacct                        profit accounting
  croponhand(farm,crop)             crop supply demand balance
  Land(farm)                        land endowment
  mincattle(farm)                   minimum number of cattle fed
  rentalLand(farm)                  maximum rented land ;
```

```
profitacct..    profit =e=
  sum(farm,
     sum(crop,-cropcost(farm,crop)*growcrops(farm,crop)
              +cropprice(farm,crop)*sellcrops(farm,crop)
              -sum(place$(not sameas(farm,place)),
                  transport*weight(crop)*movecrops(farm,place,crop)))
    -cashrent(farm)* landrent(farm)
    -1000000*artcattle(farm)
     +( cattlepric(farm)*cattlewgt(farm)
                   -cattlecost(farm)
                   -dietdat("othfeedcst")*cattleday(farm)  )
                  *feedcattle(farm) );
land(farm)..
     +cattleland* feedcattle(farm)
     +sum(crop,growcrops(farm,crop)  )
          =l= landavail(farm) +landrent(farm)$cashrent(farm);
rentalLand(farm)$rentavail(farm)..
    landrent(farm)$cashrent(farm)=l= rentavail(farm);
croponhand(farm,crop)..
 sum(place, movecrops(farm,place,crop))
           +sellcrops(farm,crop)
           +dietdat(crop)*cattleday(farm)
                       *feedcattle(farm)
  =l= cropyield(farm,crop)* growcrops(farm,crop)
    +sum(place, movecrops(place,farm,crop));
mincattle(farm)..
     +feedcattle(farm) +artcattle(farm) =g= contract(farm);
*cause artificial  to be zero
artcattle.up(farm)=0;
option lp=gamschk;
model farmmod /all/;
*add next 4 lines to create and solve model in figure2
*   rentavail(farm) =0;
*   cattlepric("farm1")=70 ;
*   sellcrops.up(farm,crop)=1000000;
*   feedcattle.up(farm)=1000000;
*add next 2 lines to create and solve model in figure3
*   artcattle.up(farm)=inf;
*   contract("farm1")= 1000;
*add next 2 lines to create and solve model in figure4
*   cropyield("farm2","corn")= cropyield("farm2","corn")*56;
*   contract(farm)=0;
solve farmmod maximizing profit using lp;
```

Footnotes

1. Note the material herein is also presented in chapter 17 written by this author in the draft book by McCarl and Spreen.

2. Problems may also include unrestricted or non-positive variables. In such cases large lower bounds on variables with negative objective function coefficients are also in order.

3. The distortion can be derived mathematically using an approach which is dual to the development in equation 8 above.

4. The artificials were added to the base model in appendix lines 48, 62, and 81; but were set to zero until needed (by line 83). Line 92 activates them.