Journal of Agricultural and Applied Economics, 30,2(December 1998):403–414

A Note on Fixing Misbehaving Mathematical Programs: Post-Optimality Procedures and GAMS-Related Software

Bruce A. McCarl

ABSTRACT

Mathematical programming formulations can yield faulty answers. Models can be unbounded, infeasible, or optimal with unrealistic answers. This article presents techniques for theory-based discovery of the cause of faulty models. The approaches are demonstrated in the context of linear programming. They have been computerized and interfaced using the General Algebraic Modeling System (GAMS), and are distributed free of charge through new GAMS versions and an online web page.

Key Words: debugging, GAMS software, mathematical programming.

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 studies have examined model verification and repair.

Some authors have addressed the topic while dealing with other issues. For example, McCarl and Apland (p. 162), in their article on validation, caution that "inconsistent data, bad coefficient placement, incomplete structure, and/or an incorrect objective function" can cause improper model results. However, they do not provide guidance as to problem detection. In an associated paper, McCarl (1984, p. 161) states, "If the model has failed [validation], discover why. . . . Repair the model and [solve it again]." Pannell (p. 222) argues, "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" (pp. 20–22), lists the following approaches without providing implementation details (pp. 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 study is to expand upon the works of McCarl and Apland, and Pannell, as well as a number of others (Greenberg 1993, 1994; Chinneck; Andersen and Andersen) by providing systematic, computerized approaches for the detection of model flaws.

The procedures here are available in computerized form linked to the General Algebraic

Bruce McCarl is a professor in the Department of Agricultural Economics, Texas A&M University.

The author gratefully acknowledges Chi-Chung Chen, Bill Nayda, Claudio Souza, and Ruby Ward for their comments during GAMSCHK development, and Harvey Greenberg for helpful comments on the exposition. This research was supported by the Texas Agricultural Experiment Station. Thanks also to Alex Meeraus and Ramesh Ramen for GAMS support.

Modeling System (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 online through the web page shown in the reference section (McCarl 1997). The GAMSCHK software has both pre- and post-solution processors to aid in discovering model problems. This article covers only 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 noted 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*) improper variables with irrelevant variables included, relevant variables excluded, or improper associated coefficient signs, placements, or magnitudes; and
(*d*) a structure that causes solvers to fail.

This article 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 (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. Chinneck developed a related concept defining an irreducible infeasible set as a set of constraints that, when any one constraint is removed, changes a model from being infeasible to being feasible. A broader term, "problematic substructure" (PS), is used here to refer to a portion of a formulation where the associated variables, constraints, and coefficients cause an improper solution.

Linear Programming Theory

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

$$(1)\quad \text{Max} \sum_j c_j X_j + \sum_i 0 \times S_i$$

$$\text{s.t.:} \quad \sum_j a_{ij} X_j + S_i = b_i \quad \forall\, i,$$

$$X_j, S_i \geq 0 \quad \forall\, i, j.$$

At optimality, given identification of a basis matrix (B) and the associated basic elements from the objective function (C_B), an expression for the shadow prices associated with the ith equation is:

$$(2)\quad u_i = (C_B B^{-1})_i,$$

while an expression for the reduced cost of the jth variable is:

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

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:

$$(4)\quad \sum_j a_{ij} X_j - b_i = -S_i \quad \forall\, i,$$

[1] Note the material provided here also is presented in chapter 17 (written by this author) in the draft textbook by McCarl and Spreen.

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

$$X_B = B^{-1}b. \tag{5}$$

Finally, following the development by McCarl and Spreen (chapter 3) and differentiating (5), we get an expression for the marginal effect of the right-hand-side changes:

$$\frac{\partial X_B}{\partial b} = B^{-1}. \tag{6}$$

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 table 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. The variable denoted *PROFIT* is maximized.[2]

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. Specifically, following the suggestion in both McCarl and Apland, and in Brooke, Kendrick, and Meeraus, large upper bounds will be assigned to bound the model. Then the solution will be investigated to find the unboundedness cause. In particular, large bounds are assigned to all variables that have desirable objective function values which are not already bounded. Given a problem with the structure in (1), then adding the constraints:

$$X_j \leq M \quad \forall\, j \text{ where } c_j > 0, \tag{7}$$

and where M is a very large number bounding the problem. The model is then solved and the solution examined for variables which equal the large bound. However, those variables only compose a portion of the PS. For example, suppose a large bound is active in the kth row. Also suppose that the right-hand side [the M-value in equation (7)] equals the right-hand side that would generate the expected solution variable values (X_B^e) plus 10^{10}. Then, via (5), the optimal value of the basic variables is:

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

where γ_{ik} is the ikth element of B^{-1}, and γ_k is the vector from the kth column of B^{-1}. Thus, the solution is composed of numbers of the magnitude expected plus γ_k coefficients times the large term in the bound. Via equation (6), the γ_{ik} give the expected change in the ith basic variable when the kth right-hand side is changed by one unit. Consequently, the γ_k vector indicates how the alteration of the large bound affects each basic variable. Thus, since γ_k is multiplied by 10^{10}, then in the computation of the optimal solution, 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 table 1 example is made unbounded by dropping the constraints on maximum rented land and entering the cattle price on Farm 1 in cents per pound, making a units error (which raises the cattle objective function to $76,415). That model is unbounded, so bounds of 10^6 are in-

[2] A GAMS formulation appears online in an expanded version of this article at http://agrinet.tamu.edu/mccarl.

Table 1. Base Example Model

		FEED CATTLE		MOVE CROPS: Farm 1 to Farm 2		MOVE CROPS: Farm 2 to Farm 1		GROW CROPS: Farm 1		GROW CROPS: Farm 2		SELL CROPS: Farm 1		SELL CROPS: Farm 2		RENT LAND			
	Profit	Farm 1	Farm 2	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Corn	Hay	Farm 1	Farm 2		RHS
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										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

Table 2. Optimal Shadow Prices for the Three Example Models

	Corn Crop on Hand		Hay Crop on Hand		Land		Min. Cattle Sold		Max. Rented Land	
Example Model	Farm 1	Farm 2	Farm 1	Farm 2	Farm 1	Farm 2	Farm 1	Farm 2	Farm 1	Farm 2
Unbounded	2.69	2.58	58.18	59.43	100	90.28	0	−35.21		
Infeasible	2.77	2.58	65.46	61.46	1E+6	100	−1E+6	−944.49	99,903	0
Unrealistic	2.40	2.29	55.00	51.00	96.49	16,160	0	0	0	16,060

cluded on the revenue-producing cattle production and crop sale variables.[3]

The resultant solution appears in tables 2 and 3. The feed cattle alternative on Farm 1 is at its large upper bound (table 3); but also notice (*a*) the large crop acreage on Farm 1, and (*b*) the large land rental on Farm 1. The PS involves cattle feeding, crop growing, and land rental—all on Farm 1. A modeler then would examine only those variables and any rows where more than one of them appears, and subsequently would identify the unboundedness cause—namely, 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 cwt, 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). The approach is as follows:

Step 1. Given an unbounded model, add large upper 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. Determine if any of the added large upper bounds are binding. If not, terminate the unboundedness finding procedure. If so, proceed 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 concerning 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 by increasing the bound value or by rescaling other model components. Second, a variant of this procedure can be utilized in GAMS where the modeler bounds only the objective function variable (the *PROFIT* variable in the example). However, such a procedure will identify only 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 identifies all variables which: (*a*) need to be bounded to preclude an unbounded solution, (*b*) are marked as unbounded when an unbounded solution is present, and (*c*) have optimal solution levels exceeding a "large value." In turn, the user can employ 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

[3] Tableaus and GAMS formulations for this and the other trial models can be found online in an expanded version of this article at the web page site given in footnote 2.

Table 3. Optimal Variable Solution Levels and Reduced Costs for Example Models

	Infeasible Example		Unbounded Example		Unrealistic Example	
Variable	Level	Reduced Cost	Level	Reduced Cost	Level	Reduced Cost
Objective	−2E+7	0	7E+10	0	1.3E+7	0
Feed Cattle Farm 1	30	0	1E+6	0	157	0
Feed Cattle Farm 2	50	0	50	0	0	−8,055
Corn Farm 1 to Farm 2	0	−0.22	0	−0.22	0	−0.22
Hay Farm 1 to Farm 2	0	−8	0	−2.75	0	−8
Corn Farm 2 to Farm 1	1,170	0	6,689	0	5.7E+6	0
Hay Farm 2 to Farm 1	22.5	0	0	−5.25	0	0
Grow Corn Farm 1	0	−1E+5	3E+5	0	0	−34
Grow Hay Farm 1	0	−1E+5	1E+5	0	21	0
Grow Corn Farm 2	24	0	67	0	800	0
Grow Hay Farm 2	12	0	8	0	0	−16,100
Sell Corn Farm 1	0	−0.37	0	1E+6	5.7E+6	0
Sell Hay Farm 1	0	−10.5	0	1E+6	0	−2.54
Sell Corn Farm 2	0	−0.61	0	1E+6	0	−0.24
Sell Hay Farm 2	0	−11.5	0	1E+6	0	−3.54
Rent Land Farm 1	200	0	9E+5	0	0	−3.51
Rent Land Farm 2	437	0	0	−9.72	700	0
Art Cattle Sales Farm 1	20	0				
Art Cattle Sales Farm 2	0	1E+6				

to a small PS makes identifying the PS cause relatively simple. Fifth, PSs can be complex. In the example, the unboundedness could have been caused by improper cattle sale weights, cattle prices, or land rental limits. Unboundedness does not necessarily occur 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. Finally, the burden involved in adding bounds when using GAMS is not high, because one can add bounds to all variables in a variable block with a single command.

Infeasible Models

Infeasibility causing PS can be found using the traditional Big "M" artificial variable approach (Hadley; Bazarra, Jarvis, and Sherali) with the added step of shadow price examination to find the full PS. The model is augmented with artificial variables added to any constraint that is not satisfied when the original problem decision variables equal 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 nonzero quantity. The artificial variables also have a large penalty cost in the objective function, which makes them highly undesirable in the optimal solution.

When the original problem is infeasible, then one or more artificials will be nonzero. Basic, nonzero artificials distort the shadow price and reduced-cost solution. Shadow prices are given by $C_B B^{-1}$. When some elements in C_B are large values associated with artificials, then some shadow prices and reduced costs will be large.[4] The constraints and nonnegativity conditions/bounds associated with these large values are the infeasibility causing PS.

Again, this is best illustrated through ex-

[4] The distortion can be derived mathematically using an approach which is dual to the development in equation (8) above.

ample. Models are infeasible because constraints are mutually inconsistent. The example is rendered infeasible by increasing the land requirement for cattle up to 10. Artificial variables also are defined for the cattle minimum sales requirements.

The solution shows large shadow prices for Farm 1 land, maximum rented land, and minimum cattle sales (table 2), as well as large crop-growing reduced costs (table 3). 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 a 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 is as follows:

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 "M"s 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. Determine if any of the artificial variables are in the basis. If not, terminate. If so, proceed 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 therein to find data errors.

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

Several comments are in order concerning 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 the entire PS in one pass. Third, in the above example, a portion 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. A PS 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. The modeler 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. Fifth, GAMSCHK will both identify constraints where artificial variables are needed, as well as equations and variables with shadow prices and reduced costs greater in absolute value than 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 still can be accomplished in a relatively few statements using the algebraic capabilities of GAMS.

While the Big "M" method is covered in many texts and papers, 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 I shadow prices. Finally, note that this approach is the dual to the large bounds approach; i.e., placing

Table 4. Farm 2 Cattle Feeding Budget for Unrealistic Optimal Case

Equation	a_{ij}	u_i	$a_{ij}u_i$
Profit Accounting	−153.20	1.00	−153.20
Corn Crop on Hand	38.48	2.29	88.04
Hay Crop on Hand	0.74	53.54	39.62
Land	0.50	16,160.40	8,080.20
Minimum Cattle	1.00	0.00	0.00
True Reduced Cost			8,054.66

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 usually can find the PS after adding artificials or bounds. More difficult cases arise when the modeler 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 termed "budgeting" here, or (*b*) allocation information using a process denoted "row summing" here.

Finding Causes of Unrealistic Solutions Through Budgeting

If the solution contains improper valuation information, then the shadow prices or reduced costs are wrong. Shadow prices are determined by the solution of the reduced-cost equations [see equation (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. Such an expansion has a row for each equation in which the *j*th variable has a nonzero 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. An example is shown in table 4.

A model with an unrealistic optimal solution is illustrated by modifying the corn yield for Farm 2 so it is in pounds rather than bushels, and zeroing the minimum cattle sale requirement. Model solution yields the information that appears in tables 2 and 3. A symptom of the model problems is found in the $8,055 reduced cost of producing Farm 2 cattle. Normal returns likely would 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 4 provides a budget for that variable.

The first five rows of table 4 contain the coefficients associated with the variable as it falls in the constraints and objective function. This cattle variable produces objective function income of $153.20 per unit, uses 38.48 units of corn, 0.74 units of hay, 0.5 units of land, and produces 1 unit of cattle to be sold. The shadow prices and product columns show the net revenue of $153.20 is counterbalanced by use of 38.48 units of 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 yields $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.

Table 5. Farm 2 Corn Production Budget for Unrealistic Optimal Case

Equation	a_{ij}	u_i	$a_{ij}u_i$
Profit Accounting	240.00	1.00	240.00
Corn Crop on Hand	−7,162.00	2.29	−16,400.00
Land (Farm 1)	1.00	96.49	16,160.00
True Reduced Cost			0.00

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 table 5.

From table 5, corn exhibits a cost of $240, a yield, and use of land. The $16,400 opportunity cost of land is shown to be mainly counterbalanced by an unrealistic 7,162 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 then would correct that problem, solve the model again, and repeat the procedure if other solution irregularities were found.

The general budgeting approach entails the following steps:

Step 1. Examine the shadow price and reduced-cost solutions to determine 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, proceed to step 3.

Step 3. Given that an unrealistic shadow price has been found, budget the basic variables that have nonzero 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 variable 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 if an a_{ij} is missing, the modeler may need to either: (*a*) investigate the constraint through row summing as discussed below, (*b*) add a missing constraint, or (*c*) add missing coefficients.

The above procedure necessitates several comments. First, budgeting always requires knowledge of the problem as well as expectations about the proper values of shadow prices and coefficients. One must understand the model to debug it. Second, the PS is identified 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 comprises 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, the modeler can request that selected variables be budgeted or that all variables which fall into particular equations be budgeted. Attention also can 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, and then accumulates to arrive at a bottom line. The procedure here is exactly analogous, with the prices used being the shadow prices derived by the solver.

Table 6. Row Sum of the Objective Function for Unrealistic Optimal Case

Variable	a_{ij}	X_j	$a_{ij}X_j$
Profit	1.00	12,867,960.00	12,867,960.00
Feed Cattle (Farm 1)	−185.00	157.14	−29,071.00
Corn Move Crops (Farm 1, Farm 2)	0.11	5,734,400.00	642,250.00
Hay Grow Crops (Farm 1)	220.00	21.43	4,714.00
Corn Grow Crops (Farm 2)	240.00	800.00	192,000.00
Corn Sell Crops (Farm 1)	−2.40	5,728,272.00	−13,747,853.00
Land Rent (Farm 2)	100.00	700.00	70,000.00
RHS Coefficient			0.00

Note: Variables which are zero are suppressed.

Finding Causes of Unrealistic Solutions Through Row Summing

The dual approach to budgeting is to look at primal allocation. When looking at primal allocation, the modeler 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 activity in the constraints. Row summing involves systematic reconstruction of constraint activity. The basic table for row summing the ith equation will have a row for each variable that falls into the ith 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, we reuse the example from the previous section on budgeting. As shown in table 6, suppose net profits of $12,867,960 are judged excessive, and a row sum of the objective function is employed to see how they arise. The objective function row sum shows the profits come largely from a $13,747,853 Farm 1 corn sale. In turn, a row sum is performed on the Farm 1 corn balance (table 7). The procedure determines that the 5,728,272 bushels sold on Farm 1 arise largely from corn transported from Farm 2. Table 8 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, proceed to step 4.

Step 4. If other variables with unreasonable

Table 7. Row Sum of Farm 1 Corn Balance for Unrealistic Optimal Case

Variable	a_{ij}	X_j	$a_{ij}X_j$
Feed Cattle (Farm 1)	39.00	157.14	6,128.00
Move Crops (Farm 1, Farm 2)	1.00	0.00	0.00
Move Crops (Farm 2, Farm 1)	−1.00	5,734,400.00	5,734,400.00
Grow Crops (Farm 1)	−130.00	0.00	0.00
Sell Crops (Farm 1)	1.00	5,728,272.00	5,728,272.00
RHS Coefficient			0.00

Table 8. Row Sum of Farm 2 Corn Balance for Unrealistic Optimal Case

Variable	a_{ij}	X_j	$a_{ij}X_j$
Feed Cattle (Farm 2)	38.48	0.00	0.00
Move Crops (Farm 2, Farm 1)	−1.00	5,734,400.00	5,734,400.00
Move Crops (Farm 1, Farm 2)	1.00	0.00	0.00
Grow Crops (Farm 2)	−7,168.00	800.00	5,734,400.00
Sell Crops (Farm 2)	1.00	0.00	0.00
RHS Coefficient			0.00

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 incidences of excessive resource use in the model.

Several comments can be made pertaining to this procedure. First, the modeler 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 misspecified coefficients.

Concluding Comments

The procedures presented here 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 also may be used in either the unbounded or 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 straightforward 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 constraints often are artificially imposed to generate an integer solution.

The procedures discussed above all are implemented in the GAMSCHK software that is distributed free of charge through the web page shown in the reference section (McCarl 1997). The procedures discussed in this article involve only post-optimality evaluations of potential structural problems; however, GAMSCHK also contains pre-optimality evaluation procedures.

References

Andersen, E.D., and K.D. Andersen. "Presolving in Linear Programming." *Mathematical Programming* 71(1995):221–45.

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

Brooke, A., D. Kendrick, and A. Meeraus. *GAMS: A User's Guide.* Version 2.25. San Francisco: Scientific Press, 1993.

Chinneck, J.W. "Feasibility and Viability." In *Recent Advances in Sensitivity Analysis and Parametric Programming,* eds., T. Gal and H.J. Greenberg, Chap. 14. Boston: 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). Online. Available HTTP: http://www-math.cudenver.edu/~hgreenbe/impsbib/impsbib.html.

———. *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for Analysis.* Boston: Kluwer Academic Publishers, 1993.

———. "How to Analyze the Results of Linear Programs, Part 4: Forcing Substructures." *Interfaces* 24,1(January/February 1994):121–30.

Hadley, G. *Linear Programming.* Reading MA: Addison-Wesley, 1962.

McCarl, B.A. *GAMSCHK USER DOCUMENTATION: A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved Using GAMS.* Dept. of Agr. Econ., Texas A&M University, 1997. Distributed through web page. Online. Available HTTP: http://agrinet.tamu.edu/mccarl.

———. "Model Validation: An Overview with Some Emphasis on Risk Models." *Rev. Mktg. and Agr. Econ.* 52,3(1984):153–73.

McCarl, B.A., and J.D. Apland. "Validation of Linear Programming Models." *S. J. Agr. Econ.* 18,2(1986):155–64.

McCarl, B.A., and T.H. Spreen. *Applied Mathematical Programming Using Algebraic Systems.* Draft textbook, Dept. of Agr. Econ., Texas A&M University, 1997. Distributed through web page. Online. Available HTTP: http://agrinet.tamu.edu/mccarl.

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