

CHAPTER V: LINEAR PROGRAMMING MODELING

CHAPTER V: LINEAR PROGRAMMING MODELING.....	1
5.1 Introduction to Algebraic Modeling.....	2
5.1.1 Defining Indices and GAMS Sets.....	2
5.1.2 Writing Equations.....	3
5.1.2.1 Unindexed Equations.....	3
5.1.2.2 Indexed Equations.....	4
5.1.3 Data Definitions.....	4
5.1.4 Entering Data Through Calculation.....	6
5.1.5 Specification of Variables.....	7
5.1.6 Equations.....	7
5.1.7 Writing Model Equations.....	7
5.1.7.1 Objective Function Equations.....	7
5.1.7.2 Constraint Equations.....	8
5.1.8 Specifying Bounds.....	9
5.1.9 Forming and Solving a Model.....	9
5.1.10 Report Writing.....	9
5.1.11 Additional GAMS Topics.....	10
5.2 GAMS and Algebraic Modeling of the Resource Allocation Problem.....	10
5.2.1 Example and GAMS Implementation.....	11
5.2.1.1 Model Solution.....	14
5.2.2 Comments.....	14
5.3 The Transportation Problem.....	14
5.3.1 Example.....	16
5.3.1.1 Model Solution.....	18
5.3.2 Comments.....	18
5.4 Diet/Feed Mix/Blending Problem.....	19
5.4.1 Example.....	21
5.4.1.1 Model Solution.....	22
5.4.2 Comments.....	22
5.5 Joint Products.....	22
5.5.1 Example.....	25
5.4.1.1 Model Solution.....	26
5.5.2 Comments.....	26
5.6 Comments on GAMS and Algebraic Modeling.....	27
5.6.1 Why Use Algebraic Modeling?.....	27
5.6.2 Why Use GAMS to Support Modeling?.....	27
5.6.3 Aid with Initial Formulation and Changes in Formulation.....	28
5.6.4 Automated Computational Tasks.....	29
5.6.5 Facilitates Documentation and Later Use of Models.....	29
5.6.6 Allows Use by Varied Personnel.....	29
5.6.7 Supported by Libraries of Applications.....	29

In this chapter we introduce and subsequently rely heavily on the algebraic approach to LP modeling, GAMS usage, duality implications and some model specification issues. The chapter begins with a presentation of the general algebraic modeling approach with GAMS.

Subsequently, we organize the presentation around commonly modeled situations. The first problem presented is the classical resource allocation problem. This is followed by the transportation and diet/feed mix/blending problems. Following these is a formulation that explicitly incorporates joint products. In presenting this material, we identify different types of variables and constraints used in building models, as well as examples of modeling assumptions used when formulating problems.

Applied duality is also treated. However, the duality material is not intended to imply that the dual of every problem must be formed when modeling. Instead, we discuss implications for the shadow prices that arise due to primal variable structures. The empirical relationship between primal and dual solutions will also be exhibited, which hopefully leaves no doubt that when solving the primal problem, the dual is simultaneously solved.

A third theme herein is communication of the empirical issues involved in LP application. This is attempted through the development of examples involving coefficient calculation and deductive steps in modeling. We will also demonstrate the link between algebraic representations of problems and empirical models. This discussion is designed to show readers the usefulness of algebraic models as a way of conceptualizing problems.

This particular chapter is not designed to stand alone. Additional formulations and algebraic/GAMS modeling techniques are presented throughout the rest of the book. Furthermore, reference to the GAMS manual (Brooke et al.) is essential to a thorough understanding of the material.

5.1 Introduction to Algebraic Modeling

Now we turn our attention to the construction of algebraic mathematical programming models using summation notation and GAMS. This section presents a brief introduction to a subset of GAMS. The GAMS manual (Brooke et al.) provides a complete treatment. This section is also supported by the material in Appendix A.

5.1.1 Defining Indices and GAMS Sets

A crucial algebraic modeling element is the identification of indices (also referred to mathematically as a subscript, or in GAMS as a SET). The compactness with which one can specify a model depends on the characterization of items with indices. However, the readability of the model depends on indices being disaggregate enough to reveal essential features.

Examples of the appropriate specification of indices appear throughout the rest of the text. Definition of indices in GAMS involves the definition of sets and set members. Namely, an index in summation notation is a SET in GAMS, and one specifies the range of the index by specifying SET membership. Examples of such set definitions are included in the following four GAMS statements.

```
SETS    I                THE INDEX I                /1,2,3,4,5/
```

CITIES	CITY NAMES	/BOSTON, PORTLAND /
YEAR	MODELS YEARS	/1970*1974/
WEST	WESTERN CITIES	/PORTLAND/;

The first item entered is the label SETS which suggests that the following material contains SET definitions. Here four sets are being defined and are called I, CITIES, YEAR and WEST. The set WEST is defined as a subset of set CITIES. Within each set definition are three components;

1. the name of the set;
2. a verbal description of what the set stands for (which keeps a record for future users and helps describe the set in the output); and
3. a list of the set members enclosed between a pair of slashes with multiple set elements separated by commas (this can also contain a definition of the set element). When a set contains consecutively numbered elements, an asterisk (as in the YEAR set definition) can be used to denote all elements in a numeric range (i.e., 1970*1974 means include 1970, 1971, 1972, 1973, and 1974). Finally, note that a semicolon is used to end the SET statement after all sets have been defined. Set elements can be defined using numbers, characters, or a mixture of both.

In GAMS, one can define subsets. A subset such as WEST (CITIES) identifies selected elements identifying such things as the western cities. Brooke, et. al. present information on more complex specifications.

5.1.2 Writing Equations

The obvious purpose of algebraic modeling is to write general algebraically based equations which encompass a wide variety of situations. Two fundamental classes of algebraic equations exist. The first class encompasses all single equations which are not defined with respect to any kind of index. A scalar calculation falls into this class. The second class involves equations which are defined according to indices. This class involves families of simultaneous equations. For example, similar equations may be defined for each location, land class, and time period.

5.1.2.1 Unindexed Equations

A summation notation version of an unindexed equation is

$$Z = \sum 3 a_i X_j$$

This equation depicts the sum across all the members of the index set j where three times a_j is multiplied times X_j and the resultant sum is placed into the value of Z. A related example is

$$Q = \sum \sum r_{ij} / y_{ij}$$

Notice in both equations that all indices are included in the sum, and the result is a scalar number. This will always be true with unindexed or scalar equations. There cannot be indices or sets within the equation which are not summed over. The GAMS statements equivalent to the above two equations are

$$Z = \text{SUM}(J, 3 * A(J) * X(J));$$

$$Q = \text{SUM} ((I,J), R(I,J)/Y(I,J));^1$$

The elements, of these GAMS equations, merit discussion. First, both equations equate a scalar to a sum. In doing this, the set or sets to be summed over are specified as the parameters to be used in forming the equation. Note, after the summation operator (SUM), the index or indices are named, followed by a comma, with the parameters separated by mathematical operators. Parentheses must be balanced. Finally, each expression is ended by a semicolon.

Although these examples are consistent with the algebraic models, they are poor examples of GAMS modeling. Namely GAMS formulations can be made much more self explanatory by utilizing set and set member names which are up to ten characters long. For example, the following two GAMS commands elaborate on those above, but are much easier to read;

```
INCOME = SUM(PRODUCT, 3* PRICE (PRODUCT) * QUANTITY (PRODUCT));
TOTALQUAN = SUM ((SOURCE, DESTINATION),
                 TOTALCOST (SOURCE, DESTINATION)/COSTPRUNIT (SOURCE, DESTINATION));
```

Here the labels help define what the equation represents.

5.1.2.2 Indexed Equations

The other class of equations commonly used in algebraic modeling involves equations defined over a family of indices. Consider for example :

$$b_i = \sum_j a_{ij} X_j \quad \text{for all } i$$

$$r_{mj} = \sum_k (f_{km} Q_{kj} + Z_{jm}) \quad \text{for all } m \text{ and } j$$

where the "for all" statement defines the family of indices for which the equations exist. Note that in these equations the index mentioned in the "for all" clause is not summed over. Rather, the "for all" clause indicates presence of these equations for each of these indices. The equivalent GAMS statements are

```
B (ITEM) = SUM (J, A (ITEM, J) * X (J)) ;
R (WHATISIT, J) = SUM (K, F (K, J, WHATISIT) * Q (I, J)) + Z (J, WHATISIT) ;
```

5.1.3 Data Definitions

When using algebraic modeling, data items are usually defined symbolically. For example, a_{ij} might be defined as the usage of the i^{th} resource per unit of the j^{th} product manufactured. In the development of an application, such items need to be defined numerically. In GAMS, raw data are entered using the SCALAR, PARAMETER and TABLE syntax.

Suppose one wishes to enter single numbers which are not referenced with respect to a SET. In this case, one uses the SCALAR syntax as follows:

```
SCALARS      LAND      LAND AVAILABLE      /100/
```

¹ Note that these GAMS statements show equations for numerical calculations, not equations for inclusion as a constraint or objective function in a linear programming model. In such cases a slightly different syntax is utilized, as defined in the next section.

HOURS	MACHINE TIME	/50/
PRICESTEEL	STEEL PRICE	/2.25/;

Again the statements contain several parts. The keyword SCALAR (or SCALARS) indicates unindexed data items are being defined. Then one enters each data item name followed by a verbal description and a numerical value enclosed in slashes. This can be followed by entries for as many scalars as necessary with the last definition followed by a semi-colon. Data are also entered using the PARAMETER syntax. Parameter statements are used to enter indexed (subscripted) data in list form. Examples are given:

PARAMETERS

PRICE (ITEM)	ITEM PRICES	/LABOR 4.00, COAL 100/
DISTANCE (I, J)	INTER CITY DISTANCES	/BOSTON.CHICAGO 20, BOSTON.CLEVELAND 15/;

This syntax must begin with the word PARAMETER (or PARAMETERS). Subsequently, the name of the data item, its indices and a description appears followed by a slash. Within the slashes the name of each data set index and their values are given. Commas separate set members and their values. This sequence repeats for each set member to be specified. In the examples, LABOR and COAL are members of the set ITEM with their associated entries in the PRICE data taking on the values 4.00 and 100 respectively. Similarly, the set I includes BOSTON and the set J includes CHICAGO and CLEVELAND. Note, the syntax in the second expression where BOSTON.CHICAGO indicates this is the data for a pair of elements (i value then j value). Finally, a semicolon indicates the PARAMETER statement end.

Yet another data entry possibility involves the TABLE syntax. This syntax is most convenient for defining multi-dimensional parameters as illustrated by:

TABLE DISTANCE (I, J) DISTANCE BETWEEN CITIES		
	CHICAGO	CLEVE
BOSTON	20	10
BALT	20	9 ;
TABLE MODECOST (I, J, MODE) INTER CITY COST OF MOVING ITEMS		
	TRUCK	RAIL
BOSTON.CHICAGO	10	5
BOSTON.CLEVE	8	7
BALT.CHICAGO	10	5

```
BALT.CLEVE          7          6          ;
```

or

```
TABLE MODECOST (I, J, MODE) INTER CITY COST OF MOVING ITEMS
      CHICAGO.TRUCK    CHICAGO.RAIL    CLEVE.TRUCK    CLEVE.RAIL
BOSTON  10            5            8            7
BALT    10            5            7            6          ;
```

The sets appear in the table structure in the order they are listed. Thus, for the above table the two dimensional item DISTANCE (I,J) is defined over the sets I and J with the names of the members of the set I appearing vertically down the left-hand side of the table while the member names for J are identified horizontally. Data values appear underneath the names of the members of the J set.

The second and third examples present alternative ways of entering a set with more than two dimensions. In the first case the first two sets are handled in the vertical column, and in the second case the last two sets are handled across the top. In both cases, the two sets are handled by identifying the member of the first indexed set followed by a period then the member name from the second set.

5.1.4 Entering Data Through Calculation

One may also calculate data. In this case, the set dependence of a parameter and a verbal description is specified without data. Then an equation is specified to define the data. For example, suppose the cost of transporting items was a known linear function of mode and distance. A procedure to implement this would be as follows:

```
PARAMETER MODECOST (I, J, MODE)    COMPUTED COST OF SHIPPING BY  MODE;
TABLE COSTUNIT (TERM, MODE)        COST DATA FOR SHIPPING MODE
      TRUCK      RAIL
      CONS      1      1.2
      PERMILE   .5     .25   ;

MODECOST (I, J, MODE) = COSTUNIT ("CONS",MODE)
      + COSTUNIT ("PERMILE",MODE)*DISTANCE (I, J);
DISPLAY MODECOST;
```

In this example the PARAMETER statement does not contain data. Rather, calculations define the variable MODECOST based on data in the tables DIST and COSTUNIT from the previous page which give distance and the dependence of transport cost on distance. After MODECOST is calculated, it is copied to the output using a DISPLAY statement. The data calculation feature is powerful as one can enter raw data and then do calculations of model parameters, and record the assumptions made in calculating the data.

One also can do simple replacements. For example, the statement

```
MODECOST(I,J,MODE) = 5.;
```

would result in having all elements of MODECOST equal to 5.

5.1.5 Specification of Variables

In linear programming models, the variables will have nonnegativity or other sign conditions. In GAMS this is identified by denoting POSITIVE VARIABLES, NEGATIVE VARIABLES, VARIABLES which are free variables, and later BINARY or INTEGER VARIABLES. For example

```
POSITIVE VARIABLES
    VOLUME (ITEM)                AMOUNT PRODUCED BY ITEM
    QUANTITY (CITY1, CITY2)      AMOUNT SHIPPED BETWEEN CITIES;
NEGATIVE VARIABLES
    LOSS                          AMOUNT OF MONEY LOST;
VARIABLES
    OBJ                          OBJECTIVE FUNCTION VALUE;
```

Note that the variable and its sign restriction are simultaneously defined. A POSITIVE specification means that the variable is restricted to be greater than or equal to zero, NEGATIVE means less than or equal to zero, and the word VARIABLE means unrestricted in sign. After these statements, the name of each variable along with its set dependence appears, and may be followed by a short description of the variable. The statement lists each variable with the particular sign restriction, followed by a semicolon. Every variable used in a GAMS model must be defined and the model must contain at least one unrestricted variable which is the item optimized.

5.1.6 Equations

The set of equations which appear in the model are specified through the EQUATIONS statement.

```
EQUATIONS
    OBJECTIVE                    OBJECTIVE FUNCTION
    AVAILABLE (RESOURCES)        RESOURCE AVAILABILITY CONSTRAINTS
    SUPPLY (PLANT)               SUPPLY AVAILABLE AT A PLANT
    LIMIT (RESOURCES, PLANT)     RESOURCES AVAILABLE BY PLANT;
```

These expressions include the name of the equation, an indication of the sets for which this equation exists and a verbal description. Entries appear for each of the model equations followed by a semicolon.

5.1.7 Writing Model Equations

After the model equations have been defined, then they are defined algebraically. The algebraic specification differs somewhat depending on whether the objective function or the constraints are being entered.

5.1.7.1 Objective Function Equations

The objective function equation is typically of the form:

$$\text{Max } \sum_j c_j x_j$$

However, GAMS forces the modeler to rewrite the equation so that the objective function expression is set equal to an unrestricted variable, and that variable is named in the solve statement. Thus we alter the problem to become

$$\begin{aligned} \text{Max } & Z \\ \text{S.t. } & Z = \sum_j c_j x_j \end{aligned}$$

where Z is a variable which is unrestricted in sign (named in the GAMS VARIABLES list). Specifying the objective function equation in GAMS requires the specification of an equation name followed by an algebraic representation. Suppose we label the objective function equation OBJECTIVEF. The resultant GAMS statement is :

$$\text{Objectivef.. } Z =E= \text{SUM} (J, c(J)*X(J));$$

Note the structure, first the equation is named followed by two periods. Then the algebraic statement appears where the unrestricted variable that is maximized is set equal to an expression for the objective function following the rules for equation writing as discussed above with one exception. That is, here the equality is written =E= . Finally, the expression is followed by a semicolon.

5.1.7.2 Constraint Equations

The constraint equations are entered in essentially the same form as the objective function. First the equation is named. In this naming any indices(sets) over which the equations are defined are named. Then this is followed by two periods and the algebraic equation for the constraint. Subsequently, an indication of the form of the inequality appears (=L= for less than or equal to; =G= for greater than or equal to; and =E= for equal to). Finally, this is following by the expression for the right-hand-side and a semi-colon. Consider for example the constraints:

$$\begin{aligned} \sum_j a_{ij} X_j & \leq b_i \quad \text{for all } i \\ \sum_k Z_{kp} & \geq d_p \quad \text{for all } p \\ \sum_j \left(\sum_i (f_{ijm} Q_{ij}) + Z_{jm} \right) & = 4 \quad \text{for all } m \\ \sum_m \left(\sum_j \left(\sum_i (f_{ijm} Q_{ij}) + Z_{jm} \right) \right) & = 0 \end{aligned}$$

where the "for all" statement defines the family of indices for which the equations exist. Suppose we label the first constraint RESOURCE indexed over the set ITEM. Similarly, we call the second constraint DEMAND indexed over PLACE, the third constraint SOMEITEM indexed over WHAT and the fourth ADDIT. The resultant GAMS statements are:

```
RESOURCE (ITEM) .. SUM (J, A (ITEM, J) *X (J)) =L= B (ITEM) ;
DEMAND (PLACE) .. SUM (K, Z (K, PLACE)) =G= D (PLACE) ;
someitem (what) .. Sum (j, sum (i, f (i, j, what) *q (i, j)) +z (j, what)) =e=4;
addit .. Sum ((m, j), sum (i, f (i, j, what) *q (i, j)) +z (j, what)) =e=0;
```

The equations follow the equation writing conventions discussed above and in Appendix A. The only exception is that one may mix constants and variables on either side of the equation. Then one could write the first equation as:

$$\text{RESOURCE (ITEM) .. } B (ITEM) =G= \text{SUM} (J, A (ITEM, J) *X (J)) ;$$

5.1.8 Specifying Bounds

Upper and lower bounds on individual variables are handled in a different fashion. For example, if the variables VOLUME and QUANTITY had been defined, and were to be bounded, then the bounds would be defined through statements like the following

```
VOLUME.UP (ITEM) = 10.;  
QUANTITY.LO (CITY1, CITY2) = MINQ (CITY1, CITY2);  
VOLUME.UP ("TREES") = 200. + LIMIT ("TREES").
```

In the first example, upper bounds are specified for the variable VOLUME across all members of the set ITEM equaling 10. In the second example, the bound is imposed through a prespecified parameter data. In the third example, the bound is calculated.

5.1.9 Forming and Solving a Model

After specifying sets, variables, data, and other appropriate input, one enters statements which define the model and invoke the solver. The MODEL definition can be of two forms. In the first form, the expression

```
MODEL RESOURCE /ALL/ ;
```

gives the model a name (RESOURCE), and specifies that ALL equations are incorporated. Alternatively, the statement

```
MODEL COSTMIN /OBJECTIVE, SUPPLY, DEMAND/;
```

gives the model name and the names of the equations to include. In this case the model name is COSTMIN and the equations included are OBJECTIVE, SUPPLY and DEMAND.

In turn, the SOLVE statement is

```
SOLVE COSTMIN USING LP MINIMIZING OBJ;
```

where the general syntax is SOLVE "modelname" USING LP MAXIMIZING [OR MINIMIZING] "objective function variable name." The objective function variable name is the variable added and set equal to the mathematical expression for the objective function one may also solve mixed integer problems by altering the using phrase to "usingmip" and nonlinear problems with "using NLP".

5.1.10 Report Writing

GAMS has useful features which allow solution results to be used in calculations to compute various items of interest or do various result summaries. One can, at any time after the SOLVE statement, use: a) a variable name followed by a .L to obtain the value of the solution for that variable; b) a variable name followed by a .M to get the reduced cost associated with that variable; c) an equation name followed by .L to get the value of the left hand side of that equation; and d) an equation name followed by .M to get the shadow price associated with the equation. Slack variables can be computed by taking an equation right hand side minus the associated .L value. The DISPLAY statement can be used to print out calculation results.

5.1.11 Additional GAMS Topics

Two additional GAMS topics are worth mentioning. These involve the use of conditional statements and the use of loops. Conditional statements are utilized to indicate that there are cases where particular items should not be defined when doing a calculation and involve a syntax form using a \$. Two examples are

```
DEMAND (REGION) $DEMANDQ (REGION) . .
SUM (SUPPLY, QUAN (SUPPLY, REGION)) =G= DEMANDQ (REGION) ;
OBJECTIVE . .      OBJ = SUM ( (I, J) $ (DIST (I, J) LT 20) ,
                           COST (I, J) *QUAN (I, J) ) ;
```

In the first case, the notation \$DEMANDQ(REGION) tells GAMS to generate this equation only if the DEMANDQ term is nonzero resulting in a DEMAND equation being generated for only those members of REGION which have nonzero demand. In the second expression, the \$(DIST(I,J) LT 20) clause indicates that one should add (i,j) pairs into the sum only if the DISTANCE between the pair is less than 20. In general, use of the \$ notation allows one to impose conditions within the model setup and data calculation procedures.

LOOP statements are also worth mentioning as they can be used to repeatedly execute a set of GAMS statements. Examples appear in the DIET example on the disc where the LOOP is utilized to vary a price and solve over and over again. Similarly, the EVPORTFO example uses a LOOP to solve for alternative risk aversion parameters.

5.2 GAMS and Algebraic Modeling of the Resource Allocation Problem

The classical LP problem involves the allocation of an endowment of scarce resources among a number of competing products so as to maximize profits. In formulating this problem algebraically let us pursue the logical steps of:

- a) index identification,
- b) variable identification,
- c) constraint identification and
- d) objective function identification.

Parameter identification is implicit in steps c and d.

The indices necessary to formulate this problem are inherent in the problem statement. Since there are "competing products" we need an index for products which we define here as j. The index j may take on the values of 1 through n indicating that there are n different competing products. The problem statement also mentions "scarce resources", so we define the index i to represent each of m possible resources (i=1,2,...,m).

Now suppose we turn our attention to variables. Two potential variables are implicit in the above statement. The first type involves variables indicating the allocation of resources and the second type is associated with the competing products. In structuring this problem we would have to know the exact amount of each resource that is used in making each product. Thus, when we know how much of a product is made, we know the quantity of each resource allocated to that product. This means the variable in this problem involves the amount of each product to be made. Suppose we define a variable X_j which is the number of units of the j^{th} product made. We may now define constraint equations. In this case, a constraint is needed for each of the scarce resources. The constraint forces resource usage across all production possibilities to be

less than or equal to the resource endowment. An algebraic statement of this restriction requires definitions of parameters for resource usage by product and the resource endowment. Let a_{ij} depict the number of units of the i^{th} resource used when producing one unit of the j^{th} product and let b_i depict the endowment of the i^{th} resource. We now can construct an algebraic form of the constraint. Namely we need to sum up the total usage of the i^{th} resource. The total usage of resource is the sum of the per unit resource usage times the number of units produced ($a_{ij} X_j$). The appropriate algebraic statement of the i^{th} constraint is

$$\sum_j a_{ij} X_j \leq b_i$$

with a constraint defined for each of the resources (i) in the problem. A condition is also needed which states only zero or positive production levels are allowed for each production possibility. $X_j \geq 0$, for all j .

Turning to the objective function, an equation for profits earned is also needed. This involves the definition of a parameter for the profit contribution per unit of the j^{th} product (c_j). Then, the algebraic statement of the objective function is the sum of the per unit profit contribution times the amount of each product made or

$$\sum_j c_j X_j$$

which is to be maximized.

Algebraically, the LP formulation of the problem is

$$\begin{aligned} \text{Max } & \sum_j c_j X_j \\ \text{s.t. } & \sum_j a_{ij} X_j \leq b_i \quad \text{for all } i \\ & X_j \geq 0 \quad \text{for all } j \end{aligned}$$

where j is indexed from 1 to n , i is indexed from 1 to m , c_j is the profit per unit of activity j , X_j is the number of units of activity j produced, a_{ij} is the number of units of resource i required per unit of product j , and b_i is the endowment of resource i .

This formulation arose early in the development of LP. While the exact problem is not stated as above, there are problems very close to it in Koopman's 1949 symposia report. The formulation explicitly appears in Dorfman's 1951 book. Kantorovich also presents an early example. Over time, this particular formulation has been widely used. In fact, it provides the first example in virtually all textbooks.

5.2.1 Example and GAMS Implementation

Suppose that E-Z Chair Makers are trying to determine how many of each of two types of chairs to produce. Further, suppose that E-Z has four categories of resources which constrain production. These involve the availability of three types of machines: 1) large lathe, 2) small lathe, and 3) chair bottom carver; as well as labor availability. Two types of chairs are produced: functional and fancy. A functional chair costs \$15 in basic materials and a fancy chair \$25. A finished functional chair sells for \$82 and a fancy chair for \$105. The resource requirements for each product are shown in Table 5.1.

The shop has flexibility in the usage of equipment. The chairs may have part of their work substituted between lathes. Labor and material costs are also affected. Data on the substitution possibilities are given in Table 5.2. Assume the availability of time is 140 hours for the small lathe, 90 hours for the large lathe, 120 hours for the chair bottom carver, and 125 hours of labor.

This problem can be cast in the format of the resource allocation problem. Six different chair/production method possibilities can be delineated. Let

X_1 = the number of functional chairs made with the normal pattern;

X_2 = the number of functional chairs made with maximum use of the small lathe;

X_3 = the number of functional chairs made with maximum use of the large lathe;

X_4 = the number of fancy chairs made with the normal pattern;

X_5 = the number of fancy chairs made with maximum use of the small lathe;

X_6 = the number of fancy chairs made with maximum use of the large lathe.

The objective function coefficients require calculation. The basic formula is that profits for the j^{th} activity (c_j) equal the revenue to the particular type of chair less the relevant base material costs, less any relevant cost increase due to lathe shifts. Thus, the c_j for X_1 is calculated by subtracting 15 from 82, yielding the entered 67. The constraints on the problem impose the availability of each of the four resources. The technical coefficients are those appearing in Tables 5.1 and 5.2.

The resultant LP model in algebraic form is

Max	67 X_1	+	66 X_2	+	66.3 X_3	+	80 X_4	+	78.5 X_5	+	78.4 X_6	
s.t.	0.8 X_1	+	1.3 X_2	+	0.2 X_3	+	1.2 X_4	+	1.7 X_5	+	0.5 X_6	# 140
	0.5 X_1	+	0.2 X_2	+	1.3 X_3	+	0.7 X_4	+	0.3 X_5	+	1.5 X_6	# 90
	0.4 X_1	+	0.4 X_2	+	0.4 X_3	+	X_4	+	X_5	+	X_6	# 120
	X_1	+	1.05 X_2	+	1.1 X_3	+	0.8 X_4	+	0.82 X_5	+	0.84 X_6	# 125
	X_1	,	X_2	,	X_3	,	X_4	,	X_5	,	X_6	∃ 0

This problem can be cast into an algebraic modeling system like GAMS in numerous ways. Two approaches are presented here. The first is faithful to the above algebraic formulation and the second is more tailored to the problem. Consider the first formulation as shown in Table 5.3 and the file called RESOURCE. The approach we employ in setting up the GAMS formulation involves a more extensive definition of both variables and constraints than in the algebraic format. The indices are defined as GAMS sets, but instead of using the short names i and j longer names are used. Thus, in statement one the set j (which stands for the different products that can be produced) is named PROCESS and the definition of the elements of j (rather than being 1-6) are mnemonics indicating which of the products is being produced and which lathe is being used. Thus, FUNCTNORM stands for producing functional chairs using the normal process while the mnemonic FANCYMXLRG stands for producing fancy chairs with the maximum use of the large lathe. Similarly, in line four, the i subscript is named RESOURCE. In turn, the parameters reflect these definitions. The availability of each resource (b_i) is specified through the RESORAVAIL parameter in lines 13-15. The per unit use of each resource by each production process (a_{ij}) is entered through the RESOURSUSE definition in lines 17-28. The

objective function profit parameter (c_j) is not directly defined but rather is calculated during model setup. The inputs to this calculation are defined in the data with the per unit prices of the chairs defined by process in PRICE lines 8-9 and the production cost defined in lines 11-12. In turn, the parameters are used in computing the objective function in line 39 where price minus production cost is computed.

The variables (X_j) are defined in lines 30-31 where the POSITIVE specification means greater than or equal to zero. Note that a variable named PROFIT is defined in line 33. This variable does not have a counterpart in the algebraic model. This unrestricted variable is required by GAMS and will be the quantity maximized. Subsequently, the two equations are specified. The equation for the objective function is not indexed by any sets and is defined in line 35. The resource constraint equation indexed by the set RESOURCE appears in line 36. In turn then, the algebraic statement of the objective function appears in lines 38-40 which sums net profits over all possible production processes by computing price minus the production cost times the level produced. The resource constraints sum resource use over all the processes and hold this less than or equal to the resource availability (lines 42-44). The final step in the GAMS implementation is to specify the model name which contains all the equations (line 46) and then a request to solve the model (line 47).

Another GAMS formulation of the problem appears in Table 5.4 and file RESOURC1. This is less faithful to the algebraic formulation, but is presented here since it shows more about GAMS. Here, the j subscript definition is broken into two parts. One of which reflects the type of CHAIR being made and the second the type of PROCESS being utilized. Thus, FUNCT.NORMAL refers to a functional chair using the normal process. In turn, this allows PRICE and BASECOST to be specified by CHAIR, but the EXTRACOST from using the additional processes needs to be specified in terms of both CHAIR and PROCESS. The resource usage matrix now has three dimensions: one for RESOURCE, one for CHAIR, and one for PROCESS. Also, the dimensions of the RESOURCUSE array are changed accordingly (line 20). The model then proceeds basically as before with the PRODUCTION variable now having two indices, CHAIR and PROCESS (line 33). The specification also means a slightly more complex expression in the objective function in which the net return to a particular chair, using a particular process is calculated as chair price minus production cost, minus extra cost. In turn, that is multiplied by the production level and summed (lines 41-42).

Examining and contrasting these formulations shows some of the power of GAMS. Namely, in both formulations, one can put in raw information such as chair prices and costs and then computations to setup the model. This leaves a record of how the numbers were calculated rather than having model coefficients computed elsewhere. In addition, the use of longer names (up to 10 characters) in specifying the model, means that the GAMS instructions can be written in sufficient detail so that they are easily understood. For example, look at the objective function equation in Table 5.4. Note, that it contains production level ties, the price of the chair minus the cost of the chair minus the extra cost by a process. This is a much more readily accessible formulation than exists in many computational approaches to linear programming (see the material below or treatments like that in Beneke and Winterboer).

5.2.1.1 Model Solution

The resultant solution under either formulation gives an optimal objective function value of 10,417.29. The optimal values of the primal variables and shadow prices are shown in Table 5.5. This solution indicates that 62.23 functional chairs and 73.02 fancy chairs should be made using the normal process, while 5.18 fancy chairs should be made using maximum use of a large lathe. This production plan exhausts small and large lathe resources as well as labor. The dual information indicates that one more hour of the small lathe is worth \$33.33, one more hour of the large lathe \$25.79, and one more hour of labor \$27.44. The reduced cost valuation information also shows, for example, that functional chair production with maximum use of a small lathe would cost \$11.30 a chair. Finally, there is excess capacity of 16.91 hours of chair bottom carving.

5.2.2 Comments

The resource allocation problem is the most fundamental LP problem. This problem has many uses; however, most uses involve slightly more complex problem structures which will be discussed in the remainder of the book.

A number of modeling assumptions are implicit in this formulation. First, the price received for production of chairs is independent of the quantity of chairs produced. The firm would receive the same price per unit whether it produced 5 or 500 million chairs. This may be unrealistic for large firms. Increased production may bring either an increasing or decreasing price.

Representation of decreasing returns is presented in the nonlinear approximation chapter and in the price endogenous modeling chapter. The integer programming chapter contains a formulation for the case where prices increase with sales.

A second assumption of the above formulation is that the fixed resource availability does not change regardless of its value. For example, the E-Z chair problem solution placed a value on labor of \$27 an hour. However, the firm may feel it can afford more labor at that price.

Consequently, one may wish to extend the model so that more of the resources become available if a sufficiently high price would be paid. This general topic will be covered under purchase activities which are introduced in the joint products problem below. It is also covered in the nonlinear approximation and price endogenous chapters.

Finally, consider an assumption that does not characterize the above formulation. Many people ordinarily regard the resource allocation problem as containing a fixed coefficient production process where there is only a single way of producing any particular product. However, we included multiple ways of producing a product in this example problem to show that a LP model may represent not only one way of producing a product, but also different ways involving different combinations of inputs.

5.3 The Transportation Problem

The second problem covered is the transportation problem. This problem involves the shipment of a homogeneous product from a number of supply locations to a number of demand locations. This problem was originally formulated by Kantorovich in 1939 and Hitchcock in 1941. Koopmans (1949) restated the model and spurred research in the area.

Setting this problem up algebraically requires definition of indices for: a) the supply points which we will designate as i , and b) the demand locations which we will designate as j . In turn, the variables indicate the quantity shipped from each supply location to each demand location. We define this set of variables as X_{ij} .

There are three general types of constraints, one allowing only nonnegative shipments, one limiting shipments from each supply point to existing supply and the third imposing a minimum demand requirement at each demand location. Definition of the supply constraint requires specification of the parameter s_i which gives the supply available at point i , as well as the formation of an expression requiring the sum of outgoing shipments from the i^{th} supply point to all possible destinations (j) to not exceed s_i . Algebraically this expression is

$$\sum_j X_{ij} \leq s_i$$

Definition of the demand constraint requires specification of the demand quantity d_j required at demand point j , as well as the formation of an expression summing incoming shipments to the j^{th} demand point from all possible supply points (i). Algebraically this yields

$$\sum_i X_{ij} \geq d_j.$$

Finally, the objective function depicts minimization of total cost across all possible shipment routes. This involves definition of a parameter c_{ij} which depicts the cost of shipping one unit from supply point i to demand point j . In turn, the algebraic formulation of the objective function is the first equation in the composite formulation below.

$$\begin{aligned} \text{Min} \quad & \sum_i \sum_j c_{ij} X_{ij} \\ \text{s.t.} \quad & \sum_j X_{ij} \leq s_i \quad \text{for all } i \\ & \sum_i X_{ij} \geq d_j \quad \text{for all } j \\ & X_{ij} \geq 0 \quad \text{for all } i, j \end{aligned}$$

This particular problem is a cost minimization problem rather than a profit maximization problem. The transportation variables (X_{ij}) belong to the general class of transformation variables. Such variables transform the characteristics of a good in terms of form, time, and/or place characteristics. In this case, the transportation variables transform the place characteristics of the good by transporting it from one location to another. The supply constraints are classical resource availability constraints. However the demand constraint imposes a minimum level and constitutes a minimum requirement constraint.

Suppose we address duality before turning to GAMS and an example. The dual of the transportation problem establishes imputed values (shadow prices) for supply at the shipping points and demand at the consumption points. These values are established so that the difference between the value of demand and the cost of supply is maximized. The dual problem is given by²

² This formulation follows after the supply equation has been multiplied through by -1 to transform it to a greater-than constraint.

$$\begin{aligned}
\text{Max} \quad & - \sum_i s_i U_i + \sum_j d_j V_j \\
\text{s.t.} \quad & - U_i + V_j \leq c_{ij} \quad \text{for all } i, j \\
& U_i, V_j \geq 0 \quad \text{for all } i, j
\end{aligned}$$

where U_i is the marginal value of one unit available at supply point i , and V_j is the marginal value of one unit demanded at demand point j .

The dual problem is best explained by first examining the constraints. The constraint associated with X_{ij} requires the value at the demand point (V_j) to be less than or equal to the cost of moving the good from point i to point j (c_{ij}) plus the value at the supply point (U_i). Consequently, the model requires the marginal value of supply at the supply point plus the transportation cost to be no smaller than the value at the demand point. This also requires the differences in the shadow prices between demand and supply points to be no greater than the transport cost. This requirement would arise in a highly competitive market. Namely if someone went into the transportation business quoting delivery and product acquisition prices, one could not charge the demanders of the good more than the price paid for a good plus the cost of moving it, or competitors would enter the market taking away business. This also shows the general nature of the dual constraint imposed by a primal transformation variable. Namely, a restriction is imposed on the difference in the shadow prices. The dual objective function maximizes profits which equal the difference between the value of the product at the demand points ($d_j V_j$) and the cost at the supply points ($s_i U_i$).

5.3.1 Example

ABC Company has three plants which serve four demand markets. The plants are in New York, Chicago and Los Angeles. The demand markets are in Miami, Houston, Minneapolis and Portland. The quantity available at each supply point and the quantity required at each demand market are

Supply Available		Demand Required	
New York	100	Miami	30
Chicago	75	Houston	75
Los Angeles	90	Minneapolis	90
		Portland	50

The assumed distances between cities are

	Miami	Houston	Minneapolis	Portland
New York	3	7	6	23
Chicago	9	11	3	13
Los Angeles	17	6	13	7

Assume that the firm has discovered that the cost of moving goods is related to distance (D) by the formula -- $\text{Cost} = 5 + 5D$. Given these distances, the transportation costs are

	Miami	Houston	Minneapolis	Portland
New York	20	40	35	120
Chicago	50	60	20	70
Los Angeles	90	35	70	40

The above data allow formulation of an LP transportation problem. Let i denote the supply points where $i=1$ denotes New York, $i=2$ Chicago, and $i=3$ Los Angeles. Let j represent the demand points where $j=1$ denotes Miami, $j=2$ Houston, $j=3$ Minneapolis, and $j=4$ Portland. Next define X_{ij} as the quantity shipped from city i to city j ; e.g., X_{23} stands for the quantity shipped from Chicago to Minneapolis. A formulation of this problem is given in Table 5.6. This formulation may also be presented in a more compact format as

	20	40	35	120	50	60	20	70	90	35	70	40	Minimize	
	1	1	1	1									#	100
					1	1	1	1					#	75
									1	1	1	1	#	90
	+1				+1				+1				∃	30
		+1				+1				+1			∃	75
			+1				+1				+1		∃	90
				+1				+1				+1	∃	50

This shows the common incidence in LP formulations of sparsity. Although there is room for 84 coefficients in the body of the constraint matrix only 24 of these are non-zero. Thus, the problem is only 24/84ths dense, revealing a large number of zeros within the body of the matrix. The right hand sides illustrate endowments of supply through the first three constraints, and minimum requirements in the next four constraints. The variables involve resource usage at the supply point and resource supply at the demand point. These activities transform the place utility of the good from the supply point to the demand point.

The GAMS implementation is presented in Table 5.7 and in the file TRANSPORT. Two sets are defined (lines 1-4) one identifies the supply plants, the other defines the demand markets. Subsequently, supply availability and demand requirements are specified through the parameter statements in lines 6-10. The distance between the plants is specified in lines 12-17. Following this, unit transport costs are computed by first defining the parameter for cost in line 20, and then expressing the formula of \$5.00 fixed cost plus \$5.00 times the distance in line 21. Next, the nonnegative shipment variables are specified (X_{ij}). The unrestricted variable called TCOST equals the objective function value. Then, three equations are specified, one equates the

in the above model. First, transportation costs are assumed to be known and independent of volume. Second, supply and demand are assumed to be known and independent of the price charged for the product. Third, there is unlimited capacity to ship across any particular transportation route. Fourth, the problem deals with a single commodity or an unchanging mix of multiple commodities.

These assumptions have spawned many extensions, including for example, the transshipment problem (Orden), wherein transshipment through intermediate cities is permitted. Another extension allows the quantity supplied and demanded to depend on price. This problem is called a spatial equilibrium model (Takayama and Judge(1973)) and is covered in the price endogenous chapter. Problems also have been formulated with capacitated transportation routes where simple upper bounds are placed on the shipment from a supply point to a demand point (i.e., X_{ij} less than or equal to UL_{ij}). These problems are generally in the purview of network theory (Bazaraa et. al., Kennington). Multi-commodity transportation problems have also been formulated (Kennington). Cost/volume relationships have been included as in the warehouse location model in the second integer programming chapter. Finally, the objective function may be defined as containing more than just transportation costs. Ordinarily one thinks of the problem wherein the c_{ij} is the cost of transporting goods from supply point i to demand point j . However, modelers such as Barnett et al. (1984) have included the supply cost, so the overall objective function then involves minimizing delivered cost. Also the transport cost may be defined as the demand price minus the transport cost minus the supply price, thereby converting the problem into a profit maximization problem.

The transport model has interesting solution properties. The constraint matrix possesses the mathematical property of unimodularity (see Bazaraa et al. (1984) for explanation) which causes the solution to be integer-valued when the right hand sides are integers. Also, when the sum of the supplies equals the sum of the demands, then the problem is degenerate, and the dual will have multiple optimal solutions. That is, the supply and demand prices will not be unique. The transportation problem has been the subject of considerable research and application. The research has led to special solution approaches including the Ford and Fulkerson out of kilter algorithm, and specializations of the primal simplex algorithm. Discussion of such topics can be found in Dantzig (1951), Bradley et al., and Glover et al. (1974). It is worthwhile to point out that the algorithms allow the problem to be solved at a factor of 100 times faster than the use of the general simplex method. Readers attempting to solve transportation problems should consult the network literature for efficient solution procedures.

There have been many applications of different versions of the transportation problem. For example, it has been used to study the effect of railroad regulatory policy (Baumel et al.), grain subterminal facility location (Hilger et al.), and port location (Barnett et al.(1984)). The assignment and/or contract award problems are transport problems which arose early in the development of LP (see the assignment and contract awards sections in Riley and Gass). There also are related formulations such as the caterer problem.

5.4 Diet/Feed Mix/Blending Problem

One of the earliest LP formulations was the diet problem along with the associated feed mix and blending variants. The diet context involves composing a minimum cost diet from a set of available ingredients while maintaining nutritional characteristics within certain bounds.

Usually, a total dietary volume constraint is also present. Stigler studied the problem before LP was developed. However, he noted that tastes and preferences cause a disparity between observed and minimum cost human diets. Basically the human diet form of this problem takes a largely expository role with few applications. However, Waugh applied LP to the livestock feed formulation problem, and it has become one of the most widely used LP applications. The formulation has also been applied to the blending of ice cream, sausage, and petroleum (Riley and Gass).

The basic model is formulated as follows. Define index (i) representing the nutritional characteristics which must fall within certain limits (protein, calories, etc.). Define index (j) which represents the types of feedstuffs available from which the diet can be composed. Then define a variable (F_j) which represents how much of each feedstuff is used in the diet. The constraints of the problem include the normal nonnegativity restrictions plus three additional constraint types: one for the minimum requirements by nutrient, one for the maximum requirements by nutrient and one for the total volume of the diet. In setting up the nutrient based constraints parameters are needed which tell how much of each nutrient is present in each feedstuff as well as the dietary minimum and maximum requirements for that nutrient. Thus, let a_{ij} be the amount of the i^{th} nutrient present in one unit of the j^{th} feed ingredient; and let UL_i and LL_i be the maximum and minimum amount of the i^{th} nutrient in the diet. Then the nutrient constraints are formed by summing the nutrients generated from each feedstuff ($a_{ij}F_j$) and requiring these to exceed the dietary minimum and/or be less than the maximum. The resultant constraints are

$$\sum_j a_{ij}F_j \leq UL_i$$

$$\sum_j a_{ij}F_j \geq LL_i$$

A constraint is also needed that requires the ingredients in the diet equal the required weight of the diet. Assuming that the weight of the formulated diet and the feedstuffs are the same, this requirement can be written as

$$\sum_j F_j = 1.$$

Finally an objective function must be defined. This involves definition of a parameter for feedstuff

cost (c_j) and an equation which sums the total diet cost across all the feedstuffs, i.e.,

$$\sum_j c_j F_j$$

The resulting LP formulation is

$$\begin{aligned} \text{Min} \quad & \sum_j c_j F_j \\ \text{s.t.} \quad & \sum_j a_{ij} F_j \leq UL_i \quad \text{for all } i \\ & \sum_j a_{ij} F_j \geq LL_i \quad \text{for all } i \\ & \sum_j F_j = 1 \\ & F_j \geq 0 \quad \text{for all } j \end{aligned}$$

This formulation depicts a cost minimization problem. The F_j activities provide an example of purchase variables, depicting the purchase of one unit of feed ingredients at an exogenously specified price which, in turn, provides the nutrient characteristics in the mixed diet. The constraints are in the form of resource limits and minimum requirements.

The dual of this problem contains variables giving the marginal value of the nutrient upper and lower limits, as well as the value of the overall volume constraint. The dual is

$$\begin{aligned} \text{Max} \quad & - \sum_i \gamma_i UL_i + \sum_i \beta_i LL_i + \alpha \\ \text{s.t.} \quad & - \sum_i \gamma_i a_{ij} + \sum_i \beta_i a_{ij} + \alpha \leq c_j \quad \text{for all } j \\ & \gamma_i, \beta_i \geq 0 \quad \text{for all } i \\ & \alpha \quad \text{unrestricted} \end{aligned}$$

The dual variables are γ_i , the marginal value of the i^{th} nutrient upper limit constraint; β_i , the marginal value of the i^{th} nutrient lower limit constraint; and α , the marginal value of the total volume constraint. The dual constraints require that the imputed cost of each ingredient plus the cost of the volume constraint to be no greater than the cost of any ingredient. The variables γ_i and β_i are nonnegative, but α is unrestricted in sign. Since the primal constraints impose nutrient requirements, the dual objective function demonstrates how the costs of the diet are allocated to individual nutritional requirements. For further economic interpretation of the dual objective function, see Thomas et al.

5.4.1 Example

Suppose we use the model in a setting where we value a potential new ingredient in the diet. Suppose that cattle feeding involves lower and upper limits on net energy, digestible protein, fat, vitamin A, calcium, salt and phosphorus. Further, suppose the feed ingredients available are corn, hay, soybeans, urea, dical phosphate, salt and vitamin A. In addition, a new product, potato slurry, is available. One kilogram of the feed is to be mixed. The costs of the ingredients per kilogram (excluding slurry) are shown in Table 5.11. The nutrient requirements are given in Table 5.12.

The nutrient requirements give the minimum and maximum amounts of each nutrient in one kilogram of feed. Thus, there must be between 0.071 and 0.130 kg of digestible protein in one kg of feed. The volume of feed mixed must equal one kilogram. The nutrient compositions of one kg of each potential feed are shown in Table 5.13. The formulation is shown in Table 5.14. The reader should also note that the potato slurry activity has been inserted with a price of P. This price will be varied.

First, suppose we specify the price of potato slurry to be one cent per kilogram, then the GAMS formulation is given in Table 5.15 and the file DIET. Three sets are involved, the first two are equivalent to the subscripts for ingredients and nutrients. The third one is one for convenience in data input and simply tells whether a minimum or maximum is being defined. In turn, the cost, minimum, maximum, and the nutrient content of each of the feedstuffs are defined. The variables identify the use of each feedstuff and a total cost variable for the objective function. The nutrient upper and lower limits, nutrients and the weight constraint are defined in lines 39-45. Then the model solve statements appear. In addition, there are instructions which appear between the ontext and offtext statements that run the model repeatedly through a LOOP

statement to generate a series of related solutions for increasing potato slurry prices. This section also contains an example of report writing. Namely, 30 slurry price scenarios are defined, and then a set is defined where the results of the 30 price runs are kept. The solution printout is suppressed (by the OPTION statement) and only the summary result is reported. Inside the LOOP a calculation sets the ingredient cost equal to the previous price plus a half cent for each scenario. In turn, the model is solved for each price level, with slurry use and shadow price recorded in the table. After the loop is finished, the table with all the results is displayed. Note the report writing features in line 55 where the .L parameter on the end of the feed use variable indicates that the optimal value of the FEED variable for slurry is recorded in a table.

5.4.1.1 Model Solution

The solution at a slurry price of 0.01 yields an objective value of 0.021 with the variable and constraint solutions as shown in Table 5.16. The optimal solution indicates that the least cost feed ration is 95.6 percent slurry, 0.1 percent vitamin A, 1.5 percent salt, 0.2 percent dicalcium phosphate, 1.4 percent urea, 1.1 percent soybeans and 0.1 percent hay. The shadow prices indicate that the binding constraints are the net energy, vitamin A, salt, and calcium minimums along with the phosphorous maximum and the weight constraint. The shadow price on the weight constraint is 1.08 cents.

The results using the code in lines 49-57 show that the above solution would hold for prices of potato slurry between 0 and +11 cents. A second ration with 87% slurry would hold for prices between roughly 11 and 12 cents. A ration with 18% slurry would hold for prices around 12.5 cents, and for any price greater than 13 cents no potato slurry would be used. The resultant demand schedule for potato slurry is graphed in Figure 5.1.

The empirical dual formulation is given in Table 5.17 and the file DUALDIET. This formulation, with P equal to 0.01, leads to a dual solution with the same objective function value as in the primal. The optimal dual solution is shown in Table 5.18. The dual is degenerate.

5.4.2 Comments

There are three assumptions within the feed formulation problem. First, the nutrient requirements are assumed constant and independent of the final product (e.g., livestock) price. Second, the quality of each feed ingredient is known. Third, the diet is assumed to depend on only feed price and nutrients.

The diet problem is widely used, especially in formulating feed rations. Animal scientists use the term "ration-balancing", and several software programs have been specifically developed to determine least cost rations. LP models of the ration problem were used by Prevatt, et al., to evaluate fed cattle production in Florida and by Thomas et al. (1992) who examined nutrient values in dairy production. Other references and discussions are given in Dorfman et al., Ashour and Anderson, and Beneke and Winterboer.

5.5 Joint Products

Many applied LP models involve production of joint products. An example would be a petroleum cracking operation where production yields multiple products such as oil and naphtha. Other examples include dairy production where production yields both milk and calves, or

forestry processing where trees yield sawdust and multiple types of sawn lumber. Here, we present a formulation explicitly dealing with joint products.

Suppose we articulate the problem as follows. Consider a profit maximizing firm who wishes to maximize total profits derived from the available production possibilities. Furthermore, suppose each of the production possibilities yields multiple products, uses some inputs with a fixed market price, and uses some resources that are available in fixed quantity.

Formulation of this model requires indices which depict the products which could be produced (p), the production possibilities (j), the fixed price inputs purchased (k), and the resources which are available in fixed quantity (m). Three types of variables need to be defined. The first (X_p) defines the total quantity of the p^{th} product sold; the second (Y_j) identifies the quantity of j^{th} production possibility utilized; and the third (Z_k) is the amount of the k^{th} input purchased. Other than nonnegativity, three types of constraints are needed. The first relates the quantity sold of each product to the quantity yielded by production. Algebraic specification requires definition of a parameter (q_{pj}) which gives the yield of each product (p) by each production possibility. The expression

$$X_p - \sum_j q_{pj} Y_j \leq 0$$

is a supply demand balance. Here demand, in the form of sales of p^{th} product, is required to be less than or equal to supply which is the amount generated across the production alternatives. Further, since production and sales are endogenous, this is written as sales minus production and is less than or equal to zero.

The second type of constraint relates the quantity purchased of each fixed price input to the quantity utilized by the production activities. The parameter r_{kj} gives the use of the k^{th} input by the j^{th} production possibility. In turn, the constraint sums up total fixed price input usage and equates it to purchases as follows:

$$\sum_j r_{kj} Y_j - Z_k \leq 0.$$

This constraint is another example of a supply demand balance where the endogenous demand in this case, use of the k^{th} input, is required to be less than or equal to the endogenous supply which is the amount purchased.

The third type of constraint is a classical resource availability constraint which insures that the quantity used of each fixed quantity input does not exceed the resource endowment.

Specification requires definition of parameters for the resource endowment (b_m) and resource use when the production possibility is utilized. The constraint which restricts total resource usage across all possibilities is

$$\sum_j s_{mj} Y_j \leq b_m$$

where s_{mj} is the use of the m^{th} resource by Y_j .

For the objective function, an expression is needed for total profits. To algebraically expressed the profits require parameters for the sales price (c_p), the input purchase cost (e_k), and any other production costs associated with production (d_j). Then the objective function can be written as

$$\sum_p c_p X_p - \sum_j d_j Y_j - \sum_k e_k Z_k$$

The individual terms do not reflect the profit contribution of each variable in an accounting sense, rather this occurs across the total model. Thus, the production variable term (d_j) does not include either the price of the products sold or the cost of all the inputs purchased, but these components are included by terms on the sales and purchase variables. The resultant composite joint products model is

$$\begin{aligned} \text{Max} \quad & \sum_p c_p X_p - \sum_j d_j Y_j - \sum_k e_k Z_k \\ \text{s.t.} \quad & X_p - \sum_j q_{pj} Y_j \leq 0 \quad \text{for all } p \\ & \sum_j r_{kj} Y_j - Z_k \leq 0 \quad \text{for all } k \\ & \sum_j s_{mj} Y_j \leq b_m \quad \text{for all } m \\ & X_p, Y_j, Z_k \geq 0 \quad \text{for all } p, j, k \end{aligned}$$

Several features of this formulation are worth mention. First, note the explicit joint product relationships. When activity Y_j is produced, a mix of joint outputs (q_{pj} , $p=1, 2, \dots$) is produced while simultaneously consuming the variable inputs both directly priced in the objective function (d_j) and explicitly included in constraints (r_{kj}), along with the fixed inputs (s_{mj}). Thus, we have a multi-factor, multi-product production relationship.

Another feature of this problem involves the types of variables and constraints which are used. The variables X_p are sales variables which sell the available quantities of the outputs. The variables Z_k are purchase variables which supply the inputs utilized in the production process. The variables Y_j are production variables. In this case, the production variables show production explicitly in the matrix, and the product is sold through another activity. The first two constraints are supply-demand balances. The reason for the use of inequalities in supply/demand balances is discussed in Chapter VII. The last constraint is a resource endowment.

The dual to this LP problem is

$$\begin{aligned} \text{Min} \quad & \sum_m U_m b_m \\ \text{s.t.} \quad & V_p \geq c_p \quad \text{for all } p \\ & -\sum_p V_p q_{pj} + \sum_k W_k r_{kj} + \sum_m U_m s_{mj} \geq -d_j \quad \text{for all } j \\ & -W_k \geq -e_k \quad \text{for all } k \\ & V_p, W_k, U_m \geq 0 \quad \text{for all } p, k, m \end{aligned}$$

where V_p is the marginal value of product p ; W_k the marginal cost of the k^{th} variable input; and U_m is the marginal value of the m^{th} fixed resource.

The dual objective function minimizes total marginal value of the fixed resources. The first constraint insures that the marginal value of each product is at least as great as the sales price. This relationship demonstrates the dual implication of a sales variable; namely, a lower bound is imposed on the shadow price for the commodity sold. The third constraint insures that the marginal value of each input is no more than its purchase price and reflects the dual implication of a purchase variable, i.e., an upper bound is imposed on the shadow price of the item which can be purchased. The second constraint is allied with the production variable and insures that the

total value of the products yielded by a process is less than or equal to the value of the inputs used in production, where the input valuation involves the direct costs in the objective function plus the imputed costs due to variable and fixed inputs.

5.5.1 Example

Consider a farm which produces both wheat and wheat straw using seven production processes. The basic data for these production processes are given in Table 5.19. The production process involves the joint production of wheat and straw using land, seed and fertilizer. The relevant prices are wheat - \$4.00 per bushel, wheat straw - \$.50 per bale, seed - \$.20/lb., and fertilizer - \$2.00 per kilogram. Also there is a \$5 per acre production cost for each of the processes and the farm has 500 acres. This leads to the empirical formulation

$$\begin{array}{l}
 \text{Max} \quad 4X_1 + .5X_2 - 5Y_1 - 5Y_2 - 5Y_3 - 5Y_4 - 5Y_5 - 5Y_6 - 5Y_7 - 2Z_1 - .2Z_2 \\
 \text{s.t.} \quad X_1 \qquad \qquad - 30Y_1 - 50Y_2 - 65Y_3 - 75Y_4 - 80Y_5 - 80Y_6 - 75Y_7 \qquad \qquad \# \ 0 \\
 \qquad \qquad X_2 - 10Y_1 - 17Y_2 - 22Y_3 - 26Y_4 - 29Y_5 - 31Y_6 - 32Y_7 \qquad \qquad \# \ 0 \\
 \qquad \qquad \qquad \qquad + 5Y_2 + 10Y_3 + 15Y_4 + 20Y_5 + 25Y_6 + 30Y_7 - Z_1 \qquad \qquad \# \ 0 \\
 \qquad \qquad \qquad \qquad 10Y_1 + 10Y_2 + 10Y_3 + 10Y_4 + 10Y_5 + 10Y_6 + 10Y_7 \qquad - Z_2 \# \ 0 \\
 \qquad \qquad \qquad \qquad Y_1 + Y_2 + Y_3 + Y_4 + Y_5 + Y_6 + Y_7 \qquad \qquad \qquad \# \ 500 \\
 X_1, X_2, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Z_1, Z_2 \geq 0
 \end{array}$$

The variables X_1 and X_2 are the sales of wheat and straw, respectively. The variables Y_1 through Y_7 are the production of wheat via the seven different input usage/output production possibilities. The variables Z_1 and Z_2 are purchases of fertilizer and seed, respectively. The first two constraints balance product sale with production; the third and fourth constraints require purchases of seed and fertilizer when production is undertaken. The fifth constraint limits land availability. Note that the Y variables produce the multiple products using multiple inputs. This problem exhibits seven different ways of producing the two products. Each of these seven different ways uses a different input combination. Also note that by increasing the fertilizer input one obtains decreasing returns in terms of the production of wheat and straw.

The GAMS formulation for this problem is given in Table 5.20 and the file JOINT. This formulation relies on four sets. One for the products; one for the purchased inputs; one for the fixed inputs; and one for the possible production processes. In turn, parameters are specified for the product prices, input costs, the production process costs, and fixed resource availability. These data are entered conventionally except for the constant production cost which is entered algebraically in line 11.

Variables and constraints are defined as associated with the above algebraic formulation, with the net income variable specified to equal the objective function. Similarly, the objective function, yield balance, input balance and available resource constraints are defined. Then the model is defined indicating all constraints and solved.

5.4.1.1 Model Solution

The solution to this problem is shown in Table 5.21. The solution implies that 40,000 bushels of wheat and 14,500 bales of straw are produced by 500 acres of the fifth production possibility using 10,000 kilograms of fertilizer and 5,000 lbs. of seed. The reduced cost information shows a \$169.50 cost for the first production possibility if undertaken. Under this production pattern, the marginal value of land is \$287.50. The shadow prices on the first four rows are the sale and purchase prices of the various outputs and inputs depicted in those rows.

The dual of the joint product example is given in Table 5.22. Note the action of the constraints. The first two insure that the shadow prices associated with the balance row for each product are at least as great as the product prices. This is a clear condition for optimality. If the shadow price fell below this, it would be rational to sell more of the product. The last two constraints insure that the values of the inputs are less than or equal to their purchase prices. If the input became worth more than its purchase price, it would make sense to purchase more until the value fell to the purchase price. The middle constraints, under re-arrangement, insure that the total imputed value of all the products from any one production method is less than or equal to the value of the fertilizer, seed and land used arising through the terms involving W_1 , W_2 and U . The optimal solution to the dual problem is given in Table 5.23 and corresponds exactly to the optimal primal solution.

5.5.2 Comments

The joint products problem illustrates: 1) the proper handling of joint products and 2) production variables where the returns from production are not collapsed into the objective function but explicitly appear in the constraints.

The formulation also illustrates the possible complexity of LP. In this case product balance constraints are incorporated in a model along with resource constraints. Also note that X_1 and X_2 , gives the sum of total output, and that Z_1 and Z_2 give the sum of total input usage on the farm which may be convenient for model interpretation. It is possible to exclude the X and Z variables by adjusting the objective function coefficients on Y so they reflect the value of the products (Y) less the costs of the inputs (Z). The larger formulation used above may be superior if model use demands easy alteration of prices or summary accounting on total production. This is discussed in the Purposeful Modeling section of the Toward Proper Modeling chapter.

Another observation is the action of the primal variables as reflected in the dual. Note that the purchase activities provide upper bounds for the dual variables associated with the output balance constraint, while the sales activities provide lower bounds for the dual variables associated with the input balance constraints, and the production activities provide relationships between the dual variables relating to inputs and outputs.

Joint product formulations have a relatively long history. It is difficult to cite many exact applications; rather such a structure is common and implicit in many models throughout the literature. Hildreth and Reiter presented an early joint products example, Beneke and Winterboer provide examples of joint products, (e.g. p. 46), and many other examples are available.

5.6 Comments on GAMS and Algebraic Modeling

It is beneficial to examine the advantages and disadvantages of this book's orientation toward algebraic modeling and GAMS usage. This is done in this section.

5.6.1 Why Use Algebraic Modeling?

Algebraic modeling refers to the formulation of optimization problems in summation notation. Why is algebraic modeling desirable and what are its inherent shortcomings? The major advantages involve the ability to concisely state problems in an abstract general fashion, largely independent of the data and exact application context while making general duality implication statements. One can produce an algebraic formulation independent of the problem size which initially can be used to consider moving products from two plants to two regions and later to consider moving products from fifty plants to a hundred regions. Such an approach is particularly useful in the textbook presentation herein. Furthermore, familiarity with algebraic modeling allows the use of formulations from the literature.

Nevertheless, these advantages are accompanied by several disadvantages. Algebraic modeling and summation notation are difficult for some users. Some people will always desire to deal with the exact problem context, not an abstract general formulation. This does lead to a strategy most modelers use when employing algebraic modeling. Namely, algebraic modeling exercises are usually supported by small tableau based sample models which capture problem essence and serve as an aid in algebraic model formulation. Dual model setup and interpretation can also be difficult. Empirically based duals are usually easier to construct than algebraic duals.

Algebraic modeling can be quite cumbersome when dealing with small, one time problems. In such cases, the generality of the presentation may not be worth the effort. Clearly, the gains to using algebraic modeling rise with problem size. When a modeler deals with large problems, the algebraic statement is probably the only thing that is thoroughly understood. Often the numerical formulation has grown out of control. Another problem when using algebraic modeling is that certain tableau based formulation structures are difficult to express algebraically. This is illustrated by the sequencing example of Chapter 7.

Algebraic modeling is not the tool of choice for small, infrequently solved problems, but rather it is best employed for medium or large sized models (more than 30 rows and/or columns) although the GAMS report writing and computation features may make it attractive for small models. In such cases, algebraic modeling can allow the modeler to gain general insights into the problem. Furthermore, coupling algebraic modeling with a modeling system like GAMS permits gains in modeling efficiency and efficacy.

5.6.2 Why Use GAMS to Support Modeling?

Now, why use GAMS? GAMS generally requires algebraic modeling and thus, has many of algebraic modeling's advantages and disadvantages. However, GAMS is easier to tailor to the problem, so its use introduces additional advantages and overcomes some of the disadvantages. The crucial point when considering whether to use GAMS involves a comparison with traditional linear programming formulation techniques. Here we compare GAMS usage with: a) direct specification of equation-based tableaus as used in LINDO (Schrage); b) direct specification of

full tableaus as in McMillan; and c) MPS-based input as in the stand alone version of MINOS (Murtaugh and Saunders).

First, let us deal with data input. Given, the resource allocation problem, alternative formulations for GAMS, a tableau-based approach as in McMillan, a LINDO input stream (Schrage (1985)), a MPS input stream (Murtaugh and Saunders(1987)) and an alternative simplified GAMS input stream appear in Table 5.24.

Several features appear in these input streams. First, the algebraic GAMS is the most complex. Second, GAMS can be used in a much simpler fashion becoming competitive with LINDO. Third, GAMS is more readable than the tableau or MPS. Fourth, the versatility of GAMS is shown by the fact that the structure of the resource allocation problem from the earlier example is maintained with a different data set (in Panel a).

Now let us deal with GAMS disadvantages. Specification of a small model almost always takes a larger file in GAMS than when using traditional equation or tableau-based methods. Namely, the LINDO or McMillan code specification of the resource allocation example takes three to five lines of input, whereas the GAMS specification takes the better part of a page. However, GAMS can be used simplistically yielding almost the same as LINDO (Panel d). Furthermore, a higher level of computer skill is required with GAMS as a batch input file, a compiler and a 250 page users guide are involved. However GAMS is generally easier to use than MPS-based methods, as an inspection of the treatment in Beneke and Winterboer or Orchard-Hays reveals.

There are also model formulation characteristics which constitute disadvantages. GAMS generally requires everything to be written algebraically which makes some potential users uncomfortable. Also some problem structures are difficult to write algebraically (i.e., see the sequencing problem in chapter VII).

Finally, there are shortcomings due to a lack of capabilities in GAMS. GAMS does not contain a set of well-developed model debugging aids as are present in many MPS-based installations. One can add custom diagnostic aids only with difficulty (The GAMS IO Library document explains how to interface FORTRAN, PASCAL and C programs through a special programming job). It is also difficult to construct an extremely problem-tailored computer implementation as can be done using FORTRAN-based computer implementation (i.e., see McCarl et al.(1977) or McCarl and Nuthall).

5.6.3 Aid with Initial Formulation and Changes in Formulation

GAMS aids both in initially formulating and subsequently revising formulations. GAMS facilitates specification and debugging of an initial formulation by allowing the modeler to begin with a small data set, then after verifying correctness expand to a much broader context. For example, one could initially specify a small transportation model with a few suppliers and demanders. Then after that model is debugged one could respecify the problem to encompass fifty shipping origins and two hundred destinations without needing to change the algebraic model. Second, GAMS makes it easy to alter the model. Historically large models have been difficult to modify. Namely when using traditional MPS-based modeling approaches it is difficult to add new constraints or variables. On the other hand, GAMS allows one to add model features much more simply. Generally, modelers do not try to make a complete formulation the first time around, rather one starts with a small formulation and then adds structural features as demanded by the analyst. GAMS also enforces consistent modeling, allowing models to be transferred between

problem contexts (i.e., code from the transport example is used in the warehouse location example in the second integer programming chapter).

5.6.4 Automated Computational Tasks

Many of the tasks that would traditionally have required a computer programmer are automated. As such, GAMS automatically does coefficient calculation; checks the formulation for obvious flaws; chooses the solver; formats the programming problem to meet the exact requirements of the solver; causes the solver to execute the job; saves and submits the advanced basis when doing related solutions; and permits usage of the solution for report writing. Also GAMS verifies the correctness of the algebraic model statements and allows empirical verification. Furthermore, since GAMS has been implemented on machines ranging from PCs to workstations to CRAY super computers, it allows portability of a model formulation between computer systems. Switching solvers is also very simple requiring changing a solver option statement or changing from using LP to using NLP. Links to spreadsheets have also been developed. Finally, GAMS does not require programming using languages like C or FORTRAN as does traditional MPS-based matrix generators and report writers.

5.6.5 Facilitates Documentation and Later Use of Models

One of the other convenient features of GAMS is its self-documenting nature. Perhaps the largest benefit GAMS has over traditional modeling techniques is that modelers can use longer variable, equation and index names as well as comments, data definitions etc., allowing a more adequate and readable problem description. Model structure, assumptions, and any calculation procedures used in the report writing are documented as a byproduct of the modeling exercise in a self-contained file. Comment statements can be inserted by placing an asterisk in column one, followed by text identifying data sources or particular assumptions being used (i.e., in some of the authors' models, comments identify data source publication and page). Under such circumstances GAMS allows either the original author or others to alter the model structure and update data.

5.6.6 Allows Use by Varied Personnel

Modeling personnel are often rare. For example, in international development contexts, detailed GAMS applications have been set-up by modeling experts but subsequently, the model is utilized by policy-makers with minimal, if any, assistance from the modeling experts. Often, given proper internal documentation and a few instructions, clerical labor and nontechnical problem analysts can handle an analysis.

5.6.7 Supported by Libraries of Applications

GAMS is rapidly becoming a defacto industry standard. Many examples are provided by the model library, Thompson and Thore; Kendrick; Zenios and their material.

References

- Ashour, S. and C. Anderson. "Linear Programming Analysis of a Cattle Feedlot." In H.M. Salkin and J. Saha (eds.). Studies in Linear Programming. New York: North Holland Co., 1975.
- Barnett, D., J. Binkley, and B. McCarl. "Port Elevator Capacity and National and World Grain Shipments." Western Journal of Agricultural Economics. 9(1984):77-89.
- Baumel, C.P., J.J. Miller, and T.P. Drinka. "The Economics of Upgrading Seventy-one Branch Rail Lines in Iowa." American Journal of Agricultural Economics. 59(1977):61-70.
- Beneke, R.R. and R. Winterboer. Linear Programming Applications to Agriculture. Ames: The Iowa State University Press, 1973.
- Bradley, S.P., A.C. Hax and T.L. Magnanti. Applied Mathematical Programming. Reading, MA: Addison-Wesley Publishing Company, Inc., 1977.
- Brooke, A., D. Kendrick and A. Meeraus. GAMS: A User's Guide. Boyd and Fraser Publishers, Version 2.25.
- Dantzig, G.B. "Application of the Simplex Method to a Transportation Problem." In T.C. Koopmans (ed.), Activity Analysis of Production and Allocation. New York: John Wiley and Sons, Inc., 1951, pp. 359-373.
- Dorfman, R. Application of Linear Programming to the Theory of the Firm. Berkeley: University of California Press, 1951.
- Dorfman, R., P.A. Samuelson, and R.M. Solow. Linear Programming and Economic Analysis. New York: McGraw-Hill Book Company, Inc., 1958.
- Ford, L.R., Jr. and D.R. Fulkerson. Flows in Networks. Princeton, NJ: Princeton University Press, 1962.
- Glover, F., D. Karney, and D. Klingman. "Implementation and Computational Comparisons of Primal, Dual and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems." Networks. 4(1974):191-212.
- Hildreth, C.G. and S. Reiter. "On the Choice of a Crop Rotation Plan." in T.J. Koopmans (ed.) Activity Analysis of Production and Allocation. Cowles Commission Monograph 13. New York: John Wiley and Sons, 1951.
- Hilger, D., B. McCarl, and J. Uhrig. "Facilities Location: The Case of Grain Subterminals." American Journal of Agricultural Economics. 59:(1977):674-681.
- Hitchcock, F.L. "The Distribution of a Product from Several Sources to Numerous Localities." Journal of Mathematical Physics. 20(1941):244-230.
- Kantorovich, L.V. "Mathematical Methods in the Organization and Planning of Production." Publication House of the Leningrad State University, 1939. 68 pp. Translated in Management Science. 6(1960):366-422.
- Kendrick, D.A. Models for Analyzing Comparative Advantage. Norwell, MA: Kluwer Academic Publishers, 1990.
- Kennington, J.L. "A Survey of Linear Cost Multicommodity Network Flows." Operations Research. 26(1978):209-236.
- Koopmans, T.C. "Optimum Utilization of the Transportation System." Proceedings of the International Statistical Conferences. (Vol. 5 reprinted as Supplement to Econometrica Vol. 17, 1949). Washington, DC, 1947.
- McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.
- McCarl, B.A. and P. Nuthall. "Linear Programming for Repeated Use in the Analysis of Agricultural Systems." Agricultural Systems. 8,1(1982):17-39.

- McMillan, C. Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines. New York: John Wiley and Sons, Inc., 1970.
- Murtaugh, B.A. and M.A. Saunders. "MINOS 5.1 User's Guide." Report SOL 83-20R, December 1983, revised January 1987, Stanford University.
- Orchard-Hays, W. Advanced Linear Programming Computing Techniques. New York: McGraw-Hill, 1968.
- Orden, A. "The Transshipment Problem." Management Science. 2(1956):276-85.
- Prevatt, J.W., B.E. Melton, T.H. Spreen, W.K. Mathis. "The Economics of Carcass Beef Production: and Appraisal of Florida's Feedlot Potential." Southern Journal of Agricultural Economics. 10,2(1978):49-55.
- Riley, V. and S.I. Gass. Linear Programming and Associated Techniques. Published for Operations Research Office. Baltimore, MD: The Johns Hopkins Press, 1958.
- Salkin, H.M. and J. Saha. Studies in Linear Programming. New York: American Elsevier, 1975.
- Schrage, L. Linear, Integer, and Quadratic Programming with LINDO: User's Manual. Second Edition. The Scientific Press, 1985.
- Stigler, G.J. "The Cost of Subsistence." Journal of Farm Economics. 27(1945):303-314.
- Takayama, T. and G. Judge. Studies in Economic Planning over Space and Time. New York: North Holland Co., 1973.
- Thomas, C., M.A. DeLorenzo, D.K. Beede, and T.H. Spreen. "Predicting Daily Feed Costs for Dairy Management Models." Journal of Dairy Science. Forthcoming, 1992.
- Thompson, Gerald L. and Sten Thore. Computational Economics: Economic Modeling with Optimization Software. San Francisco, CA: The Scientific Press, 1992.
- Waugh, F.V. "The Minimum-Cost Dairy Feed." Journal of Farm Economics. 33(1951):299-310.
- Williams, H.P. Model Building in Mathematical Programming. 2nd Edition. Chichester; New York: John Wiley & Sons, 1985.
- Zenios, S.A., editor, 1992, "Financial Optimization," Cambridge University Press.

Table 5.1. Resource Requirements for the E-Z Chair Makers for the Normal Pattern

	Hours of Use per Chair Type	
	Functional	Fancy
Small Lathe	0.8	1.2
Large Lathe	0.5	0.7
Chair Bottom Carver	0.4	1.0
Labor	1.0	0.8

Table 5.2. Resource Requirements and Increased Costs for Alternative Methods to Produce Functional and Fancy Chairs in Hours of Use per Chair and Dollars

	Maximum Use of Small Lathe		Maximum Use of Large Lathe	
	Functional	Fancy	Functional	Fancy
Small Lathe	1.30	1.70	0.20	0.50
Large Lathe	0.20	0.30	1.30	1.50
Chair Bottom Carver	0.40	1.00	0.40	1.00
Labor	1.05	0.82	1.10	0.84
Cost Increase	\$1.00	\$1.50	\$0.70	\$1.60

Table 5.3. First GAMS Formulation of Resource Allocation Example

```

1  SET      PROCESS      TYPES OF PRODUCTION PROCESSES
2              /FUNCTNORM , FUNCTMXSML , FUNCTMXLRG
3              ,FANCYNORM , FANCYMXSML , FANCYMXLRG/
4          RESOURCE      TYPES OF RESOURCES
5              /SMLLATHE, LRGLATHE, CARVER, LABOR/ ;
6
7  PARAMETER PRICE (PROCESS)      PRODUCT PRICES BY PROCESS
8              /FUNCTNORM 82, FUNCTMXSML 82, FUNCTMXLRG 82
9              ,FANCYNORM 105, FANCYMXSML 105, FANCYMXLRG 105/
10         PRODCOST (PROCESS)      COST BY PROCESS
11         /FUNCTNORM 15, FUNCTMXSML 16 , FUNCTMXLRG 15.7
12         ,FANCYNORM 25, FANCYMXSML 26.5, FANCYMXLRG 26.6/
13         RESORAVAIL (RESOURCE)    RESOURCE AVAILABILITY
14         /SMLLATHE 140, LRGLATHE 90,
15         CARVER 120, LABOR 125/;
16
17  TABLE RESOURUSE (RESOURCE, PROCESS) RESOURCE USAGE
18
19         FUNCTNORM  FUNCTMXSML  FUNCTMXLRG
20  SMLLATHE          0.80      1.30      0.20
21  LRGLATHE          0.50      0.20      1.30
22  CARVER            0.40      0.40      0.40
23  LABOR             1.00      1.05      1.10
24  +
25         FANCYNORM  FANCYMXSML  FANCYMXLRG
26  SMLLATHE          1.20      1.70      0.50
27  LRGLATHE          0.70      0.30      1.50
28  CARVER            1.00      1.00      1.00
29  LABOR             0.80      0.82      0.84;
30
31  POSITIVE VARIABLES
32         PRODUCTION (PROCESS)    ITEMS PRODUCED BY PROCESS;
33  VARIABLES
34         PROFIT                    TOTALPROFIT;
35  EQUATIONS
36         OBJT                        OBJECTIVE FUNCTION ( PROFIT )
37         AVAILABLE (RESOURCE)        RESOURCES AVAILABLE ;
38  OBJT..  PROFIT =E=
39         SUM (PROCESS, (PRICE (PROCESS) -PRODCOST (PROCESS))
40         * PRODUCTION (PROCESS)) ;
41
42  AVAILABLE (RESOURCE) ..
43         SUM (PROCESS, RESOURUSE (RESOURCE, PROCESS) *PRODUCTION (PROCESS))
44         =L= RESORAVAIL (RESOURCE) ;
45
46  MODEL RESALLOC /ALL/;
47  SOLVE RESALLOC USING LP MAXIMIZING PROFIT;

```

Table 5.4. Second GAMS Formulation of Resource Allocation Example

```

1  SET      CHAIRS      TYPES OF CHAIRS /FUNCTIONAL, FANCY/
2          PROCESS      TYPES OF PRODUCTION PROCESSES
3              /NORMAL , MXSMLLATHE , MXLRGLATHE/

```

```

4          RESOURCE          TYPES OF RESOURCES
5                               /SMLLATHE,LRGLATHE,CARVER,LABOR/ ;
6
7  PARAMETER PRICE (CHAIRS)      PRODUCT PRICES BY PROCESS
8                               /FUNCTIONAL 82, FANCY 105/
9          COST (CHAIRS)         BASE COST BY CHAIR
10                              /FUNCTIONAL 15, FANCY 25/
11         EXTRACOST (CHAIRS,PROCESS) EXTRA COST BY PROCESS
12                              / FUNCTIONAL.MXSMLLATHE 1.0 ,
13                              FUNCTIONAL.MXLRGLATHE 0.7
14                              ,FANCY.      MXSMLLATHE 1.5,
15                              FANCY.      MXLRGLATHE 1.6/
16         RESORAVAIL (RESOURCE) RESOURCE AVAILABILITY
17                              /SMLLATHE 140, LRGLATHE 90,
18                              CARVER 120, LABOR 125/;
19
20  TABLE RESOURUSE (RESOURCE,CHAIRS,PROCESS) RESOURCE USAGE
21         FUNCTIONAL.NORMAL FUNCTIONAL.MXSMLLATHEFUNCTIONAL.MXLRGLATHE
22  SMLLATHE          0.80          1.30          0.20
23  LRGLATHE          0.50          0.20          1.30
24  CARVER            0.40          0.40          0.40
25  LABOR             1.00          1.05          1.10
26  +                FANCY.NORMAL      FANCY.MXSMLLATHE      FANCY.MXLRGLATHE
27  SMLLATHE          1.20          1.70          0.50
28  LRGLATHE          0.70          0.30          1.50
29  CARVER            1.00          1.00          1.00
30  LABOR             0.80          0.82          0.84 ;
31
32  POSITIVE VARIABLES
33         PRODUCTION (CHAIRS,PROCESS) ITEMS PRODUCED BY PROCESS;
34  VARIABLES
35         PROFIT          TOTAL PROFIT;
36  EQUATIONS
37         OBJT          OBJECTIVE FUNCTION ( PROFIT )
38         AVAILABLE (RESOURCE) RESOURCES AVAILABLE ;
39
40  OBJT..  PROFIT =E= SUM ( (CHAIRS,PROCESS) ,
41                (PRICE (CHAIRS) -COST (CHAIRS) -EXTRACOST (CHAIRS,PROCESS) )
42                * PRODUCTION (CHAIRS,PROCESS) ) ;
43
44  AVAILABLE (RESOURCE) ..
45         SUM ( (CHAIRS,PROCESS) ,
46         RESOURUSE (RESOURCE,CHAIRS,PROCESS) *PRODUCTION (CHAIRS,PROCESS) )
47         =L= RESORAVAIL (RESOURCE) ;
48
49  MODEL RESALLOC /ALL/;
50  SOLVE RESALLOC USING LP MAXIMIZING PROFIT;

```

Table 5.5. Optimal Solution to the E-Z Chair Makers Problem

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X_1	62.23	0	1	0	33.33
X_2	0	-11.30	2	0	25.79
X_3	0	-4.08	3	16.91	0
X_4	73.02	0	4	0	27.44
X_5	0	-8.40			
X_6	5.18	0			

Table 5.6. Formulation of Transportation Example

Min	$20X_{11}$	+	$40X_{12}$	+	$35X_{13}$	+	$120X_{14}$	+	$50X_{21}$	+	$60X_{22}$	+	$20X_{23}$	+	$70X_{24}$	+	$90X_{31}$	+	$35X_{32}$	+	$70X_{33}$	+	$40X_{34}$	
s.t.	X_{11}	+	X_{12}	+	X_{13}	+	X_{14}																# 100	
								X_{21}	+	X_{22}	+	X_{23}	+	X_{24}										# 75

$$\begin{array}{rcccccccc}
 & & & & & & X_{31} & + & X_{32} & + & X_{33} & + & X_{34} & \# & 90 \\
 X_{11} & & & + & X_{21} & & + & X_{31} & & & & & & & \exists & 30 \\
 & X_{12} & & & & + & X_{22} & & + & X_{32} & & & & & \exists & 75 \\
 & & X_{13} & & & + & X_{23} & & & + & X_{33} & & & & \exists & 90 \\
 & & & X_{14} & & & + & X_{24} & & & & + & X_{34} & & \exists & 50 \\
 & & & & X_{ij} \geq 0 & & & & & & & & & & &
 \end{array}$$

Table 5.7. GAMS Statement of Transportation Example

```

1 SETS PLANT PLANT LOCATIONS
2 /NEWYORK, CHICAGO, LOSANGLS/
3 MARKET DEMAND MARKETS
4 /MIAMI, HOUSTON, MINEPLIS, PORTLAND/
5
6 PARAMETERS SUPPLY(PLANT) QUANTITY AVAILABLE AT EACH PLANT
7 /NEWYORK 100, CHICAGO 75, LOSANGLS 90/
8 DEMAND(MARKET) QUANTITY REQUIRED BY DEMAND MARKET
9 /MIAMI 30, HOUSTON 75,
10 MINEPLIS 90, PORTLAND 50/;
11
12 TABLE DISTANCE(PLANT,MARKET) DISTANCE FROM EACH PLANT TO EACH MARKET
13
14 MIAMI HOUSTON MINEPLIS PORTLAND
15 NEWYORK 3 7 6 23
16 CHICAGO 9 11 3 13
17 LOSANGLS 17 6 13 7;
18
19
20 PARAMETER COST(PLANT,MARKET) CALCULATED COST OF MOVING GOODS ;
21 COST(PLANT,MARKET) = 5 + 5 * DISTANCE(PLANT,MARKET) ;
22
23 POSITIVE VARIABLES
24 SHIPMENTS(PLANT,MARKET) AMOUNT SHIPPED OVER A TRANSPORT ROUTE;
25 VARIABLES
26 TCOST TOTAL COST OF SHIPPING OVER ALL ROUTES;
27 EQUATIONS
28 TCOSTEQ TOTAL COST ACCOUNTING EQUATION
29 SUPPLYEQ(PLANT) LIMIT ON SUPPLY AVAILABLE AT A PLANT
30 DEMANDEQ(MARKET) MINIMUM REQUIREMENT AT A DEMAND MARKET;
31
32 TCOSTEQ.. TCOST =E= SUM((PLANT,MARKET), SHIPMENTS(PLANT,MARKET) *
33 COST(PLANT,MARKET)) ;
34
35 SUPPLYEQ(PLANT) ..SUM(MARKET, SHIPMENTS(PLANT,MARKET)) =L= SUPPLY(PLANT) ;
36
37 DEMANDEQ(MARKET) ..SUM(PLANT, SHIPMENTS(PLANT,MARKET)) =G= DEMAND(MARKET) ;
38
39 MODEL TRANSPORT /ALL/;
40 SOLVE TRANSPORT USING LP MINIMIZING TCOST;

```

Table 5.8. Optimal Solution to the ABC Company Problem

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X ₁₁	30	0	1	20	0
X ₁₂	35	0	2	0	-15
X ₁₃	15	0	3	0	-5
X ₁₄	0	75	4	0	20
X ₂₁	0	45	5	0	40
X ₂₂	0	35	6	0	35
X ₂₃	75	0	7	0	45
X ₂₄	0	40			
X ₃₁	0	75			
X ₃₂	40	0			
X ₃₃	0	40			
X ₃₄	50	0			

Table 5.9. Optimal Shipping Pattern for the ABC Company

Origin	Destination							
	Miami		Houston		Minneapolis		Portland	
	Units	Variable	Units	Variable	Units	Variable	Units	Variable
New York	30	X ₁₁	35	X ₁₂	15	X ₁₃		
Chicago					75	X ₂₃		
Los Angeles			40	X ₃₂			50	X ₃₄

Table 5.10. Optimal Dual Solution to the ABC Company Problem

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
U ₁	0	-20	1	-20	30
U ₂	15	0	2	40	35
U ₃	5	0	3	35	15
V ₁	20	0	4	45	0
V ₂	40	0	5	5	0
V ₃	35	0	6	25	0
V ₄	45	0	7	20	75
			8	30	0
			9	15	0
			10	35	40
			11	30	0
			12	40	50

Table 5.11. Ingredient Costs for Diet Problem Example per kg

Corn	\$0.133
Dical	\$0.498
Alfalfa hay	\$0.077
Salt	\$0.110
Soybeans	\$0.300
Vitamin A	\$0.286
Urea	\$0.332

Table 5.12. Required Nutrient Characteristics per Kilogram of Mixed Feed

Nutrient	Unit	Minimum Amount	Maximum Amount
Net energy	Mega calories	1.34351	--
Digestible protein	Kilograms	0.071	0.13
Fat	Kilograms	--	0.05
Vitamin A	International Units	2200	--
Salt	Kilograms	0.015	0.02
Calcium	Kilograms	0.0025	0.01
Phosphorus	Kilograms	0.0035	0.012
Weight	Kilograms	1	1

Table 5.13. Nutrient Content per Kilogram of Feeds

Characteristic	Corn	Hay	Soybean	Urea	Dical Phosphate	Salt	Vitamin A Concentrate	Potato Slurry
Net energy	1.48	0.49	1.29					1.39
Digestible protein	0.075	0.127	0.438	2.62				0.032
Fat	0.0357	0.022	0.013					0.009
Vitamin A	600	50880	80				2204600	
Salt						1		
Calcium	0.0002	0.0125	0.0036		0.2313			0.002
Phosphorus	0.0035	0.0023	0.0075	0.68	0.1865			0.0024

Table 5.14. Primal Formulation of Feed Problem									
	Corn	Hay	Soybean	Urea	Dical	Salt	Vitamin A	Slurry	
Min	.133X _C	+ .077X _H	+ .3X _{SB}	+ .332X _{Ur}	+ .498X _d	+ .110X _{SLT}	+ .286X _{VA}	+ PX _{SL}	
s.t.	.075X _C	+ .127X _H	+ .438X _{SB}	+ 2.62X _{Ur}				+ .032X _{SL}	# .13
Max	.0357X _C	+ .022X _H	+ .013X _{SB}					+ .009X _{SL}	# .05
Nut						X _{SLT}			# .02
	.0002X _C	+ .0125X _H	+ .0036X _{SB}		+ .2313X _d			+ .002X _{SL}	# .01
	.0035X _C	+ .0023X _H	+ .0075X _{SB}	+ .68X _{Ur}	+ .1865X _d			+ .0024X _{SL}	# .012
Min	1.48X _C	+ .49X _H	+ 1.29X _{SB}					+ 1.39X _{SL}	∃ 1.34351
	.075X _C	+ .127X _H	+ .438X _{SB}	+ 2.62X _{Ur}				+ .032X _{SL}	∃ .071
Nut	600X _C	+ 50880X _H	+ 80X _{SB}				+ 2204600X _{VA}		∃ 2200
						X _{SLT}			∃ .015
	.0002X _C	+ .0125X _H	+ .0036X _{SB}		+ .2313X _d			+ .002X _{SL}	∃ .0025
	.0035X _C	+ .0023X _H	+ .0075X _{SB}	+ .68X _{Ur}	+ .1865X _d			+ .0024X _{SL}	∃ .0035
Volume	X _C	+ X _H	+ X _{SB}	+ X _{Ur}	+ X _d	+ X _{SLT}	+ X _{VA}	+ X _{SL}	= 1

Table 5.15 GAMS Formulation of Diet Example

```

1 SET INGREDT NAMES OF THE AVAILABLE FEED INGREDIENTS
2   /CORN, HAY, SOYBEAN,UREA,DICAL,SALT,VITA,SLURRY/
3   NUTRIENT NUTRIENT REQUIREMENT CATEGORIES
4   /NETENGY, PROTEIN, FAT, VITALIM, SALTLM, CALCIUM, PHOSPHRS/
5   LIMITS TYPES OF LIMITS IMPOSED ON NUTRIENTS /MINIMUM, MAXIMUM/;
6 PARAMETER INGREDCOST(INGREDT) FEED INGREDIENT COSTS PER KG PURCHASED
7   /CORN .133, HAY .077, SOYBEAN .300, UREA .332
8   , DICAL .498,SALT .110, VITA .286, SLURRY .01/;
9 TABLE NUTREQUIRE(NUTRIENT, LIMITS) NUTRIENT REQUIREMENTS
10      MINIMUM      MAXIMUM
11 NETENGY      1.34351
12 PROTEIN      .071      .130
13 FAT          0          .05
14 VITALIM      2200
15 SALTLM       .015      .02
16 CALCIUM      .0025     .0100
17 PHOSPHRS     .0035     .0120;
18 TABLE CONTENT(NUTRIENT, INGREDT) NUTRIENT CONTENTS PER KG OF FEED
19      CORN HAY SOYBEAN UREA DICAL SALT VITA SLURRY
20 NETENGY   1.48 .49  1.29                1.39
21 PROTEIN   .075 .127 .438      2.62      0.032
22 FAT       .0357 .022 .013                0.009
23 VITALIM   600 50880 80                2204600
24 SALTLM    1
25 CALCIUM   .0002 .0125 .0036      .2313      .002
26 PHOSPHRS .0035 .0023 .0075 .68 .1865      .0024;
27
28 POSITIVE VARIABLES
29   FEEDUSE(INGREDT) AMOUNT OF EACH INGREDIENT USED IN MIXING FEED;
30 VARIABLES
31   COST PER KG COST OF THE MIXED FEED;
32 EQUATIONS
33   OBJT          OBJECTIVE FUNCTION ( TOTAL COST OF THE FEED )
34   MAXBD(NUTRIENT) MAXIMUM LIMITS ON EACH NUTRIENT IN THE BLENDED FEED
35   MINBD(NUTRIENT) MINIMUM LIMITS ON EACH NUTRIENT IN THE BLENDED FEED
36   WEIGHT       REQUIREMENT THAT EXACTLY ONE KG OF FEED BE PRODUCED;
37
38   OBJT.. COST =E= SUM(INGREDT,INGREDCOST(INGREDT) * FEEDUSE(INGREDT))
39   MAXBD(NUTRIENT)$NUTREQUIRE(NUTRIENT,"MAXIMUM")..
40     SUM(INGREDT,CONTENT(NUTRIENT , INGREDT) * FEEDUSE(INGREDT))
41     =L= NUTREQUIRE(NUTRIENT, "MAXIMUM");
42   MINBD(NUTRIENT)$NUTREQUIRE(NUTRIENT,"MINIMUM")..
43     SUM(INGREDT, CONTENT(NUTRIENT, INGREDT) * FEEDUSE(INGREDT))
44     =G= NUTREQUIRE(NUTRIENT, "MINIMUM");
45   WEIGHT.. SUM(INGREDT, FEEDUSE(INGREDT)) =E= 1. ;
46 MODEL FEEDING /ALL/;
47 SOLVE FEEDING USING LP MINIMIZING COST;
48
49 SET VARYPRICE PRICE SCENARIOS /1*30/
50 PARAMETER SLURR(VARYPRICE,*)
51 OPTION SOLPRINT = OFF;
52 LOOP (VARYPRICE,
53   INGREDCOST("SLURRY")= 0.01 + (ORD(VARYPRICE)-1)*0.005;
54 SOLVE FEEDING USING LP MINIMIZING COST;
55 SLURR(VARYPRICE,"SLURRY") = FEEDUSE.L("SLURRY");
56 SLURR(VARYPRICE,"PRICE") = INGREDCOST("SLURRY") ) ;
57 DISPLAY SLURR;

```

Table 5.16. Optimal Primal Solution to the Diet Example Problem

Variable	Value	Reduced Cost	Equation	Slack	Price
X _C	0	0.095	Protein L Max	0.059	0
X _H	0.001	0	Fat Max	0.041	0
X _{SB}	0.011	0	Salt Max	0.005	0
X _{Ur}	0.014	0	Calcium Max	0.007	0
X _d	0.002	0	Phosphrs	0.000	-2.207
X _{SLT}	0.015	0	Net Engy Min	0.000	0.065
X _{VA}	0.001	0	Protein Min	0.000	0.741
X _{SL}	0.956	0	Vita Lim Min	0.000	0
			Salt Lim Min	0.000	0.218
			Calcium Min	.000	4.400
			Phosphrs	0.008	0
			Weight	0.000	-0.108

Table 5.17. Dual Formulation of Feed Mix Example Problem

	(1)	(2)	(3)	(4)	(5)	∃ ₁	∃ ₂	∃ ₃	∃ ₄	∃ ₅	∃ ₆	∃
Max	-.13	-.05	-.02	-.01	-.12	+ 1.34351	+ .071	+ 2200	+ .015	+ .0025	+ .0035	+ 1
	-.075	-.0357		-.0002	-.0035	+ 1.48	+ .075	+ 600		+ .0002	+ .0035	+ 1 # .133
	-.127	-.022		-.0125	-.0023	+ .49	+ .127	+ 50880		+ .0125	+ .0023	+ 1 # .077
	-.438	-.013		-.0036	-.0075	+ 1.29	+ .438	+ 80		+ .0036	+ .0075	+ 1 # .3
	-2.62				-.68		+ 2.62				+ .68	+ 1 # .332
				-.2313	-.1865					+ .2313	+ .1865	+ 1 # .498
		- 1							+ 1			+ 1 # .110
							+ 2204600					+ 1 # .286
	-.032	-.009		-.002	-.0024	+ 1.39	+ .032			+ .002	+ .0024	+ 1 # P

Table 5.18. Optimal Solution to the Dual of the Feed Mix Problem

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
ζ_1	0	-0.059	Corn	0.095	0
ζ_2	0	-0.041	Hay	0	0.001
ζ_3	0	-0.005	Soybean	0	0.011
ζ_4	0	-0.007	Urea	0	0.014
ζ_5	2.207	0	Dical	0	0.002
\exists_1	0.065	0	Salt	0	0.015
\exists_2	0.741	0	Vita	0	0.001
\exists_3	0	0	Slurry	0	0.956
\exists_4	0.218	0			
\exists_5	4.400	0			
\exists_6	0	-0.008			
\forall	-0.108	0			

Table 5.19. Data for the Wheat and Straw Example Problem

Outputs and Inputs Per Acre								
Wheat yield in bu.	30	50	65	75	80	80	75	
Wheat straw yield/bales	10	17	22	26	29	31	32	
Fertilizer use in Kg.	0	5	10	15	20	25	30	
Seed in pounds	10	10	10	10	10	10	10	

Table 5.20. GAMS Formulation of the Joint Products Example

```
1 SET PRODUCTS LIST OF ALTERNATIVE PRODUCT /WHEAT, STRAW/
```

```

2      INPUTS      PURCHASED INPUTS      /SEED, FERT/
3      FIXED      FIXED INPUTS      /LAND/
4      PROCESS    POSSIBLE INPUT COMBINATIONS /Y1*Y7/;
5
6  PARAMETER  PRICE (PRODUCTS)  PRODUCT PRICES  /WHEAT 4.00, STRAW 0.50/
7            COST (INPUTS)    INPUT PRICES    /SEED 0.20, FERT 2.00/
8  PRODCOST (PROCESS)  PRODUCTION COSTS BY PROCESS
9      AVAILABLE (FIXED)  FIXED INPUTS AVAILABLE / LAND 500 /;
10
11      PRODCOST (PROCESS) = 5;
12
13  TABLE YIELDS (PRODUCTS, PROCESS) YIELDS FROM THE PRODUCTION POSSIBILITIES
14      Y1  Y2  Y3  Y4  Y5  Y6  Y7
15  WHEAT  30  50  65  75  80  80  75
16  STRAW  10  17  22  26  29  31  32;
17
18  TABLE USAGE (INPUTS, PROCESS) PURCHASED INPUT USAGE BY PRODUCTION POSSIBILITIES
19      Y1  Y2  Y3  Y4  Y5  Y6  Y7
20  SEED   10  10  10  10  10  10  10
21  FERT   0   5  10  15  20  25  30;
22
23  TABLE FIXUSAGE (FIXED, PROCESS) FIXED INPUT USAGE BY PRODUCTION POSSIBILITIES
24      Y1  Y2  Y3  Y4  Y5  Y6  Y7
25  LAND   1   1  1   1   1   1   1;
26
27  POSITIVE VARIABLES
28      SALES (PRODUCTS)      AMOUNT OF EACH PRODUCT SOLD
29      PRODUCTION (PROCESS)  LAND AREA GROWN WITH EACH INPUT PATTERN
30      BUY (INPUTS)          AMOUNT OF EACH INPUT PURCHASED ;
31  VARIABLES
32      NETINCOME  NET REVENUE (PROFIT);
33  EQUATIONS
34      OBJT      OBJECTIVE FUNCTION (NET REVENUE)
35      YIELDBAL (PRODUCTS)  BALANCES PRODUCT SALE WITH PRODUCTION
36      INPUTBAL (INPUTS)    BALANCE INPUT PURCHASES WITH USAGE
37      AVAIL (FIXED)        FIXED INPUT AVAILABILITY;
38
39  OBJT..      NETINCOME =E=
40              SUM (PRODUCTS , PRICE (PRODUCTS) * SALES (PRODUCTS))
41              - SUM (PROCESS , PRODCOST (PROCESS) * PRODUCTION (PROCESS))
42              - SUM (INPUTS , COST (INPUTS) * BUY (INPUTS));
43  YIELDBAL (PRODUCTS) ..
44      SUM (PROCESS, YIELDS (PRODUCTS, PROCESS) * PRODUCTION (PROCESS))
45      =G=      SALES (PRODUCTS);
46  INPUTBAL (INPUTS) ..
47      SUM (PROCESS, USAGE (INPUTS, PROCESS) * PRODUCTION (PROCESS))
48      =L=      BUY (INPUTS);
49  AVAIL (FIXED) ..
50      SUM (PROCESS, FIXUSAGE (FIXED, PROCESS) * PRODUCTION (PROCESS))
51      =L=      AVAILABLE (FIXED);
52
53  MODEL JOINT /ALL/;
54  SOLVE JOINT USING LP MAXIMIZING NETINCOME;

```

Table 5.21. Optimal Solution of the Wheat and Straw Example Problem

Objective Function Value 143,750

Variable	Value	Reduced Cost	Equation	Slack	Shadow Price
X_1	40,000	0	Wheat	0	-4

X_2	14,500	0	Straw	0	-0.5
Y_1	0	-169.50	Fertilizer	0	2
Y_2	0	-96.00	Seed	0	0.2
Y_3	0	-43.50	Land	0	287.5
Y_4	0	-11.50			
Y_5	500	0			
Y_6	0	-9.00			
Y_7	0	-38.50			
Z_1	10,000	0			
Z_2	5,000	0			

Table 5.22. Dual Formulation of Example Joint Products Problem

Min					500U
s.t.	V_1				\exists 4
		V_2			\exists .5
	$-30V_1$	$-10V_2$	$+10W_2$	$+U$	\exists -5

$$\begin{array}{rccccccc}
-50V_1 & - & 17V_2 & + & 5W_1 & + & 10W_2 & + & U & \exists & -5 \\
-65V_1 & - & 22V_2 & + & 10W_1 & + & 10W_2 & + & U & \exists & -5 \\
-75V_1 & - & 26V_2 & + & 15W_1 & + & 10W_2 & + & U & \exists & -5 \\
-80V_1 & - & 29V_2 & + & 20W_1 & + & 10W_2 & + & U & \exists & -5 \\
-80V_1 & - & 31V_2 & + & 25W_1 & + & 10W_2 & + & U & \exists & -5 \\
-75V_1 & - & 32V_2 & + & 30W_1 & + & 10W_2 & + & U & \exists & -5 \\
& & & & -W_1 & & & & & \exists & -2 \\
& & & & & & -W_2 & & & \exists & -0.2 \\
V_1 & , & V_2 & , & W_1 & , & W_2 & , & U & \exists & 0
\end{array}$$

Table 5.23. Optimal Dual Solution of the Wheat and Straw

Example Problem

Objective Function Value 143,750

V_1	Prod 1 Cost	Prod 2 Cost	Resource	Quantity
V_1	4	0	Wheat	0
V_2	0.5	0	Straw	0
W_1	2	0	Prod 1	169.5
W_2	0.2	0	Prod 2	96
U	287.5	0	Prod 3	43.5
			Prod 4	11.5
			Prod 5	0
			Prod 6	9
			Prod 7	38.5
			Fertilizer	0
			Seed	0

Table 5.24 Alternative Computer Inputs for a Model

Simple GAMS Input File

```

POSITIVE VARIABLES X1 , X2, X3
VARIABLES Z
EQUATIONS OBJ, CONSTRAIN1 , CONSTRAIN2;
OBJ.. Z =E= 3 * X1 + 2 * X2 + 0.5* X3;
CONSTRAIN1.. X1 + X2 +X3=L= 10;
CONSTRAIN2.. X1 - X2 =L= 3;
MODEL PROBLEM /ALL/;
SOLVE PROBLEM USING LP MAXIMIZING Z;

```

Lindo Input

```

MAX 3 * X1 + 2 * X2 + 0.5* X3;
ST
X1 + X2 +X3 < 10
X1 - X2 < 3
END
GO

```

Tableau Input File

5	3				
3.	2.	0.5	0.	0.	
1.	1.	1.	1.	0.	10.
1.	-1.	0.	0.	1.	3.

MPS Input File

```

NAME CH2MPS
ROWS
N R1
L R2
L R3
COLUMNS
X1 R0 3. R1 1.
X1 R3 1.
X2 R0 2. R1 1.
X2 R1 -1.
X3 R0 0.5 R1 1.
RHS
RHS1 R1 10. R1 3.
ENDDATA

```

More Complex GAMS input file

```

SET PROCESS TYPES OF PRODUCTION PROCESSES /X1,X2,X3/
RESOURCE TYPES OF RESOURCES /CONSTRAIN1,CONSTRAIN2/
PARAMETER
PRICE(PROCESS) PRODUCT PRICES BY PROCESS /X1 3,X2 2,X3 0.5/
PRODCOST(PROCESS) COST BY PROCESS /X1 0 ,X2 0, X3 0/
RESORAVAIL(RESOURCE) RESOURCE AVAILABILITY /CONSTRAIN1 10 ,CONSTRAIN2 3/
TABLE RESOURUSE(RESOURCE,PROCESS) RESOURCE USAGE
X1 X2 X3
CONSTRAIN1 1 1 1
CONSTRAIN2 1 -1
POSITIVE VARIABLES PRODUCTION(PROCESS) ITEMS PRODUCED BY PROCESS;
VARIABLES PROFIT TOTALPROFIT;
EQUATIONS OBJT OBJECTIVE FUNCTION ( PROFIT )
AVAILABLE(RESOURCE) RESOURCES AVAILABLE ;
OBJT.. PROFIT=E= SUM(PROCESS, (PRICE(PROCESS) -PRODCOST(PROCESS)) *
PRODUCTION(PROCESS)) ;
AVAILABLE(RESOURCE) .. SUM(PROCESS, RESOURUSE(RESOURCE, PROCESS)
*PRODUCTION(PROCESS)) =L= RESORAVAIL(RESOURCE);
MODEL RESALLOC /ALL/;
SOLVE RESALLOC USING LP MAXIMIZING PROFIT;

```

Figure 5.1 Demand Schedule for Potato Slurry in Feed Mix Example

