CHAPTER III SOLUTION OF LP PROBLEMS: A MATRIX ALGEBRA APPROACH

Contents

Linear programming solution has been the subject of many articles and books. Complete coverage of LP solution approaches is beyond the scope of this book and is present in many other books. However, an understanding of the basic LP solution approach and the resulting properties are of fundamental importance. Thus, we cover LP solution principles from a matrix algebra perspective demonstrating the simplex algorithm and the properties of optimal solutions. In addition, we cover several practical matters.

3.1 Matrix Formulation of the Linear Programming Problem

The matrix version of the basic LP problem can be expressed as in the equations below.

$$\text{Max} \quad CX$$

$$\text{s.t.} \quad AX \quad < \quad b$$

$$X \quad > \quad 0$$

Here the term CX is maximized where C is an 1xN vector of profit contributions and X is an Nx1 vector of decision variables. This maximization is subject to inequality constraints involving M resources so that A is an MxN matrix giving resource use coefficients by the X's, and b is an Mx1 vector of right hand side or resource endowments. We further constrain X to be non-negative in all elements.

It is common to convert the LP inequality system to equalities by adding slack variables. These variables account for the difference between the resource endowment (b) and the use of resources by the variables (AX) at no cost to the objective function. Thus, define

$$S = b - AX$$

as the vector of slack variables. Each slack variable is restricted to be nonnegative thereby insuring that resource use is always less than or equal to the resource endowment. One slack variable is added for each constraint equation. Rewriting the constraints gives

$$AX + IS = b,$$

where I is an MX M identity matrix and S is a Mx1 vector. The slack variables appear in the objective function with zero coefficients. Thus, we add an 1xM vector of zero's to the objective function and conditions constraining the slack variables to be nonnegative. The resultant augmented LP is

$$\text{MAX} \quad CX + OS$$

$$\text{s.t.} \quad AX + IS = b$$

$$X, \quad S > 0.$$

Throughout the rest of this section we redefine the X vector to contain both the original X's and the slacks. Similarly, the new C vector will contain the original C along with the zeros for the slacks, and the new A matrix will contain the original A matrix along with the identity matrix for the slacks. The resultant problem is

$$\text{MAX} \quad CX$$

$$\text{s.t.} \quad AX = b$$

$$X \geq 0$$

3.2 Solving LP's by Matrix Algebra

LP theory (Dantzig(1963); Bazarra, et al.) reveals that a solution to the LP problem will have a set of potentially nonzero variables equal in number to the number of constraints. Such a solution is called a Basic Solution and the associated variables are commonly called Basic Variables. The other variables are set to zero and are called the nonbasic variables. Once the basic variables have been chosen; the X vector may be partitioned into X_B, denoting the vector of the basic variables, and X_{NB}, denoting the vector of the nonbasic variables. Subsequently, the problem is partitioned to become

$$\text{MAX } C_B X_B + C_{NB} X_{NB}$$
$$\text{s.t. } B X_B + A_{NB} X_{NB} = b$$
$$X_B , \quad X_{NB} \geq 0.$$

The matrix B is called the Basis Matrix, containing the coefficients of the basic variables as they appear in the constraints. A_{NB} contains the coefficients of the nonbasic variables. Similarly C_B and C_{NB} are the objective function coefficients of the basic and nonbasic variables.

Now suppose we address the solution of this problem via the simplex method. The simplex solution approach relies on choosing an initial B matrix, and then interactively making improvements. Thus, we need to identify how the solution changes when we change the B matrix. First, let us look at how the basic solution variable values change. If we rewrite the constraint equation as

$$B X_B = b - A_{NB} X_{NB}.$$

Setting the nonbasic variables (X_{NB}) to zero gives

$$B X_B = b.$$

This equation may be solved by premultiplying both sides by the inverse of the basis matrix (assuming non-singularity) to obtain the solution for the basic variables,

$$B^{-1} B X_B = I X_B = B^{-1} b \quad \text{or} \quad X_B = B^{-1} b.$$

We may also examine what happens when the nonbasic variables are changed from zero. Multiply both sides of the equation including the nonbasic variables by B^{-1} giving

$$X_B = B^{-1} b - B^{-1} A_{NB} X_{NB}.$$

This expression gives the values of the basic variables in terms of the basic solution and the nonbasic variables. This is one of the two fundamental equations of LP. Writing the second term of the equation in summation form yields

$$X_B = B^{-1} b - \sum_{j \in NB} B^{-1} a_j x_j$$

where NB gives the set of nonbasic variables and aj the associated column vectors for the nonbasic variablesXj from the original A matrix. This equation shows how the values of the basic variables are altered as the value of nonbasic variables change. Namely, if all but one (X_η) of the nonbasic variables are left equal to zero then this equation becomes

$$X_B = B^{-1}b - B^{-1}a_\eta X_\eta$$

This gives a simultaneous system of equations showing how all of the basic variables are affected by changes in the value of a nonbasic variable. Furthermore, since the basic variables must remain non-negative the solution must satisfy

$$X_{Bi*} = (B^{-1}b)_{i*} - (B^{-1}a_\eta)_{i*} X_\eta = 0$$

This equation permits the derivation of a bound on the maximum amount the nonbasic variable X_η can be changed while the basic variables remain non-negative. Namely, X_η may increase until one of the basic variables becomes zero. Suppose that the first element of X_B to become zero is X_{Bi*}. Solving for X_{Bi*} gives

$$X_{Bi*} = (B^{-1}b)_{i*} - (B^{-1} a_\eta)_{i*} X_\eta = 0$$

where $(\ )_i$ denotes the ith element of the vector. Solving for X_η yields

$$X_\eta = (B^{-1}b)_{i*}/(B^{-1}a_\eta)_{i*}, \text{ where } (b^{-1}a_\eta)_{i*}$$

This shows the value of X_η which causes the i^{*th} basic variable to become zero. Now since X_η must be nonnegative then we need only consider cases in which a basic variable is decreased by increasing the nonbasic variable. This restricts attention to cases where $(B^{-1} a_\eta)_i$, is positive. Thus, to preserve non-negativity of all variables, the maximum value of X_η is

$$X_\eta = \{(B^{-1}b)_i/(B^{-1}a_\eta)_i\} \text{ for all i where } (B^{-1}a_\eta)_i > 0$$

The procedure is called the minimum ratio rule of linear programming. Given the identification of a nonbasic variable, this rule gives the maximum value the entering variable can take on. We also know that if i* is the row where the minimum is attained then the basic variable in that row will become zero. Consequently, that variable can leave the basis with X_η inserted in its place. Note, if the minimum ratio rule reveals a tie, (i.e., the same minimum ratio occurs in more than one row), then more than one basic variable reaches zero at the same time. In turn, one of the rows where the tie exists is arbitrarily chosen as i* and the new solution has at least one zero basic variable and is degenerate . Also, note that if all the coefficients of X_η are zero or negative $(B^{-1}a_\eta)_i$ -- for all i -- then this would indicate an unbounded solution, if increasing the value of the nonbasic variable increases the objective function, since the variable does not decrease the value of any basic variables.

Another question is which nonbasic variable should be increased? Resolution of this question requires consideration of the objective function. The objective function, partitioned between the basic and nonbasic variables, is given by

$$Z = C_B X_B + C_{NB} X_{NB}$$

Substituting the X_B equation (3.1) yields

$$Z = C_B(B^{-1}b - B^{-1}A_{NB}X_{NB}) + C_{NB}X_{NB}$$

or

$$Z = C_B B^{-1}b - C_B B^{-1}A_{NB}X_{NB} + C_{NB}X_{NB}$$

or

$$Z = C_B B^{-1}b - (C_B B^{-1}A_{NB} - C_{NB})X_{NB}$$

This is the second fundamental equation of linear programming. Expressing the second term in summation notation yields

$$Z = C_B B^{-1}b - \sum_{j \in NB}\left(C_B B^{-1}a_j - c_j\right)X_j$$

This expression gives both the current value of the objective function for the basic solution ($C_B B^{-1}b$ since all nonbasic X_j equal zero) and how the objective function changes given a change in the value of nonbasic variables. Namely, when changing X_η

$$Z = C_B B^{-1}b - \left(C_B B^{-1}a_\eta - c_\eta\right)X_\eta$$

Since the first term of the equation is equal to the value of the current objective function (Z), then it can be rewritten as

$$Z = \overline{Z} - \left(C_B B^{-1} a_\eta - c_\eta\right) X_\eta$$

For maximization problems, the objective value will increase for any entering nonbasic variable if its term, $C_B B^{-1} a_\eta - c_\eta$, is negative. Thus the criterion that is most commonly used to determine which variable to enter is: <u>select the nonbasic variable that increases the value of objective function the most per unit of the variable entered</u>. Namely, we choose the variable to enter as that variable X· such that the value of $C_B B^{-1} a_\eta - c_\eta$, is most negative. This is the <u>simplex criterion rule</u> of linear programming and the term $C_B B^{-1} a_\eta - c_\eta$, is called the reduced cost.

If there are no variables with negative values of $C_B B^{-1} a_\eta - c_\eta$, then the solution cannot be improved on and is optimal. However, if a variable is identified by this rule then it should be entered into the basis. Since the basis always has a number of variables equal to the number of constraints, then to put in a new variable one of the old basic variables must be removed. The variable to remove is that basic variable which becomes zero first as determined by the minimum ratio rule. This criteria guarantees the non-negativity condition is maintained providing the initial basis is non-negative. These results give the fundamental equations behind the most popular method for solving LP problems which is the simplex algorithm. (Karmarkar presents an alternative method)

3.2.1 The Simplex Algorithm

Formally, the matrix algebra version of the simplex algorithm (assuming that an initial feasible invertible basis has been established) for a maximization problem follows the steps:

1) Select an initial feasible basis B; commonly this is composed of all slack variables and is the identity matrix.
2) Calculate the Basis inverse (B^{-1}).
3) Calculate $C_B B^{-1} a_j - c_j$ for the nonbasic variables and identify the entering variable as the variable which yields the most negative value of that calculation; denote that variable as X_η if there are none, go to step 6.
4) Calculate the minimum ratio rule.
   $$\underset{i}{\text{Min}} \left(B^{-1} b\right)_i / \left(B^{-1} a_\eta\right)_i \quad \text{where} \left(B^{-1} a_\eta\right)_i > 0$$
   Denote the row where the minimum ratio occurs as row I*; if there are no rows with $(B^{-1} a_\eta)i > 0$ then go to step 7.
5) Remove the variable that is basic in row I* by replacing the variable in the η*th column of the basis matrix with column a_η and recalculate the basis inverse. Go to step 3.
6) The solution is optimal. The optimal variable values equal $B^{-1}b$ for the basic variables and zero for the nonbasic variables. The optimal reduced costs are

7) $C_B B^{-1} a_j - c_j$ (also commonly called $Z_j - c_j$). The optimal value of the objective function is $C_B B^{-1} b$. Terminate.

8) The problem is unbounded. Terminate.

3.2.2 Example

Maximize $Z = 2000 \ X_{fancy} + 1700 \ X_{fine}$

s.t.
$$X_{fancy} + X_{fine} \leq 12$$
$$25 \ X_{fancy} + 20 \ X_{fine} \leq 280$$
$$X_{fancy} \ , \ X_{fine} \geq 0$$

Now suppose we choose s_1 and s_2 to be in the initial basis. Thus, initially

$$X_B = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \qquad X_{NB} = \begin{bmatrix} X_{fancy} \\ X_{fine} \end{bmatrix}$$

$$C = \begin{bmatrix} 2000 & 1700 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 25 & 20 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 12 \\ 280 \end{bmatrix}$$

$$C_B B^{-1} A_{NB} - C_{NB} = \begin{bmatrix} -2000 & -1700 \end{bmatrix}$$

$$C_B = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad\qquad C_{NB} = \begin{bmatrix} 2000 & 17000 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad\qquad A_{NB} = \begin{bmatrix} 1 & 1 \\ 25 & 20 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Now using criterion for selecting the entering variables ($C_B B^{-1} a_{NB} - c_{NB}$):

Taking the variable associated with the most negative value (-2000) from this calculation indicates the first nonbasic variable X_{fancy}, should enter. Computation of the minimum ratio rule requires the associated $B^{-1} a_1$ and $B^{-1} b$

$$B^{-1} a_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 25 \end{bmatrix} = \begin{bmatrix} 1 \\ 25 \end{bmatrix}$$

$$X_B = B^{-1} b = \begin{bmatrix} 12 \\ 280 \end{bmatrix}.$$

Using the criterion for leaving variable

$$Min_i \left[(B^{-1}b)_i / (B^{-1}a_1)_i \right] = Min \begin{bmatrix} 12/1 \\ 280/25 \end{bmatrix} = \begin{bmatrix} 12 \\ 11.2 \end{bmatrix} = 11.2 \Rightarrow i* = 2$$

In this case, the minimum ratio occurs in row 2. Thus, we replace the second basic variable, s_2, with X_{fancy}. At this point, the new basic and nonbasic items become

$$X_B = \begin{vmatrix} S_1 \\ X_{fancy} \end{vmatrix} \qquad\qquad X_{NB} = \begin{vmatrix} X_{fine} \\ S_2 \end{vmatrix}$$

$$C_B = \begin{bmatrix} 0 & 2000 \end{bmatrix} \qquad\qquad C_{NB} = \begin{bmatrix} 1700 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 25 \end{bmatrix} \qquad\qquad A_{NB} = \begin{bmatrix} 1 & 0 \\ 20 & 1 \end{bmatrix}$$

and the new basis inverse is

$$B_1^{-1} = \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix}$$

Recomputing the reduced costs for the nonbasic variables X_{fine}, and s_2 gives

$$C_B B^{-1} A_{NB} - C_{NB} = \begin{bmatrix} 0 & 2000 \end{bmatrix} \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 20 & 1 \end{bmatrix} - \begin{bmatrix} 1700 & 0 \end{bmatrix} = \begin{bmatrix} -100 & 80 \end{bmatrix}$$

Observe that the procedure implies X_{fine} should enter this basis. The coefficients for the minimum ratio rule are

$$B_1^{-1}a_2 = \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix} \begin{bmatrix} 1 \\ 20 \end{bmatrix} = \begin{bmatrix} 1/5 \\ 4/5 \end{bmatrix}, \quad B_1^{-1}b = \begin{bmatrix} 0.8 \\ 11.2 \end{bmatrix}$$

The minimum ratio rule computation yields

$$Min \begin{bmatrix} 0.8/(1/5) \\ 11.2/(4/5) \end{bmatrix} = \begin{bmatrix} 4 \\ 14 \end{bmatrix} = 4 \Rightarrow i^* = 1$$

In the current basis, s_1 is the basic variable associated with row 1. Thus, replace s_1 with X_{fine}. The new basis vector is $[X_{fine} \ X_{fancy}]$ and the basic matrix is now

$$B = \begin{bmatrix} 1 & 1 \\ 20 & 25 \end{bmatrix}$$

In turn the basis inverse becomes

$$B^{-1} = \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix}$$

The resultant reduced costs are

$$C_B B^{-1} A_{NB} - C_{NB} = \begin{bmatrix} 1700 & 2000 \end{bmatrix} \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 500 & 60 \end{bmatrix}$$

Since all of these are greater than zero, this solution is optimal. In this optimal solution

$$X_B = \begin{bmatrix} X_{fine} \\ X_{fancy} \end{bmatrix} = B_2^{-1} b = \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} \begin{bmatrix} 12 \\ 280 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$$

$$Z = C_B X_B = 22800$$

$$C_B B^{-1} a_j - c_j = \begin{bmatrix} 0 & 0 & 500 & 60 \end{bmatrix} j \in NB$$

This method may be expanded to handle difficulties with finding the initial nonnegative basis using either the Phase I/Phase II or BIG M methods discussed below.

3.3 Solutions and Their Interpretation

LP solutions arise and are composed of a number of elements. In this section we discuss general solution interpretation, common solver solution format and contents, special solution cases and sensitivity analysis.

3.3.1 General Solution Interpretation

The two fundamental equations developed in section 3.1 may be utilized to interpret the LP solution information. The first (3.1) shows how the basic variables change as nonbasic variables are changed,

$$X_B = B^{-1} b - \sum_{j \in NB} B^{-1} a_j x_j$$

and the second (3.2) give the associated change in the objective function when a nonbasic variable is changed

$$Z = C_B B^{-1} b - \sum_{j \in NB} \left(C_B B^{-1} a_j - c_j \right) x_j,$$

Suppose we assume that an optimal basic solution has been found and that B and B^{-1} are the associated basis and basis inverse. Now suppose we consider changing the constraint right hand sides. The implications of such a change for the solution information may be explored using calculus. Differentiating the above equations with respect to the right hand side b yields

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

$$\frac{\partial Z}{\partial b} = C_B B^{-1}$$

These results indicate that $C_B B^{-1}$ is the expected rate of change in the objective function when the right hand sides are changed. The values $C_B B^{-1}$ are called the <u>shadow prices</u> and give estimates of the marginal values of the resources (later they will also be called the <u>Dual Variables</u> or <u>Dual Solution</u>). Similarly, B^{-1} gives the expected rate of change in the basic variables when resources are changed. Thus when the first right hand side is changed, the basic variables change at the rate given by the first column within the basis inverse; i.e., the first variable changes at rate $(B^{-1})_{11}$, the second at $(B^{-1})_{21}$ and so on.

Other results may be derived regarding changes in nonbasic variables. Partially differentiating the objective function equation with respect to a nonbasic variable yields

$$\frac{\partial Z}{\partial x_j} = - (C_B B^{-1} a_j - c_j) \quad j \in NB$$

This shows that the expected marginal cost of increasing a nonbasic variable equals the negative of $C_B B^{-1} a_j - c_j$, a consequence the $C_B B^{-1} a_j - c_j$ term is usually called <u>reduced cost</u>. The marginal effect of changes in the nonbasic variables on the basic variables is obtained by differentiating. This yields

$$\frac{\partial X_B}{\partial x_j} = - B^{-1} a_j \quad j \in NB$$

which shows that the marginal effect of the nonbasic variables on the basic variable is minus $B^{-1}a_j$. The B^{-1} constitutes a translation from the original resource use space (i.e., a_j) into the basic variables space and tells us how many units of each basic variable are removed with a marginal change in the nonbasic variable. We can also use these results to further interpret the $\frac{\partial Z}{\partial x_j}$ equation. The marginal revenue due to increasing a non basic variable is equal to its direct revenue (c_j the objective function coefficient) less the value of the basic variables (C_B) times the amount of the basic variables diverted ($B^{-1}a_j$). Thus, this equation takes into account both the direct effect from increasing X_j plus the substitution effect for the basic variables.

3.3.2 Examples of Solution Interpretation

This set of general interpretations may be applied to the Joe's Van example above. The appropriate mathematical expressions for each of the four items are as follows.

$$\frac{\partial Z}{\partial b} = C_B B^{-1} = \begin{bmatrix} 1700 & 2000 \end{bmatrix} \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} = \begin{bmatrix} 500 & 60 \end{bmatrix}$$

$$\frac{\partial X_B}{\partial b} = B^{-1} = \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix}$$

$$\frac{\partial Z}{\partial X_{NB}} = -(C_B B^{-1} A_{NB} - C_{NB}) = -\begin{bmatrix} 1700 & 2000 \end{bmatrix} \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} -500 & -61 \end{bmatrix}$$

$$\frac{\partial X_B}{\partial X_{NB}} = -B^{-1} a_{NB} = -\begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -5 & 1/5 \\ 4 & -1/5 \end{bmatrix}$$

The first expression, which gives the partial of Z with respect to b, tells how the objective function changes when the right hand sides change. Thus, if the capacity limit were changed upward from 12, one would expect the objective function to increase $500 per unit. Similarly if the second right hand side or the labor limit were increased upwards from 280 then one would expect a return of $60 per hour.

The second expression indicates the anticipated change in the values of the basic variables when the right hand sides are changed; the basic variables in the model are arranged with X_{fine} being first and X_{fancy} being second. The first column of the basis inverse corresponds to what happens if the van capacity right hand side is changed; whereas, the second column corresponds to what happens if the labor right hand side is changed. Thus, if capacity were expanded to 13, one would expect to produce 5 more fine vans and 4 less fancy vans. Similarly, if labor was expanded, the number of fine vans would decrease by 1/5 per unit and the number of fancy vans would increase by 1/5. The particular signs of these tradeoffs are caused by the original data. Fancy vans use more labor then fine vans. Thus, when capacity is expanded, more fine vans are made since they use labor more intensively while, if labor is increased, one makes more fancy vans.

Now let us examine the effects of changes on the objective function when the nonbasic variables are altered. In this problem we have two nonbasic variables which are the slack variables for the two resources. The effect of increasing the nonbasics is a $500 decrease if we increase slack capacity, and a $60 decrease if we increase slack labor. This is exactly the opposite of the resource values discussed above, since the consequence of increasing the slacks is the same as that of decreasing the resource endowments.

The interpretation of the basis inverse also allows us to get further information about the interpretation of the change in the objective function when the right hand sides have changed. Namely, if changing capacity causes five more fine vans to be produced (each worth $1700,

leading to a $8500 increase but, four less fancy vans worth $8000) the net effect then is a $500 increase equaling the shadow price. Similarly, the labor change causes $400 more worth of fancy vans to be produced but $340 less fine vans for a net value of $60. Overall, this shows an important property of linear programming. The optimal solution information contains information about how items substitute. This substitution information is driven by the relative uses of the constraint resources by each of the alternative activities. This is true in more complex linear programming solutions.

3.3.3 Finding Limits of Interpretation

The above interpretations only hold when the basis remains feasible and optimal. Ranging analysis is the most widely utilized tool for analyzing how much a linear program can be altered without changing the interpretation of the solution. Ranging analysis deals with the question: what is the range of values for a particular parameter for which the current solution remains optimal? Ranging analysis of right-hand-side (b_i) and objective function coefficients (c_j) is common; many computer programs available to solve LP problems have options to conduct ranging analyses although GAMS does not easily support such features (See chapter 19 for details).

3.3.3.1 Ranging Right Hand Sides

Let us study what happens if we alter the right hand side (RHS). To do this let us write the new RHS in terms of the old RHS, the size of the change and a direction of change,

$$b_{new} = b_{old} + \theta r$$

where b_{new} is the new RHS, b_{old} is the old RHS, θ is a scalar giving the magnitude of the change and r is the direction of change. Given an r vector, the resultant values for the basic variables and objective function are

$$X_B \quad = \quad B^{-1}b_{new} \quad = \quad B^{-1}(b_{old} + \theta r) \quad = \quad B^{-1}b_{old} \quad + \quad \theta B^{-1}r$$

while $C_B B^{-1}a_j - c_j$ is unchanged. The net effect is that the new solution levels are equal to the old solution levels plus $\theta B^{-1}r$. Similarly the new objective function is the old one plus $\theta C_B B^{-1}$ r. For the basis to still hold the basic variables must remain nonnegative as must the reduced costs ($C_B B^{-1}a_j - c_j$). However, since the reduced costs are unaltered we must only worry about the basic variables. Thus the condition for X_B can be written with non-negativity imposed

$$X_B \ = \ B^{-1}b_{new} \ = \ B^{-1}b_{old} + \theta B^{-1}r \ \geq \ 0$$

and merits further examination in terms of the admissible value of θ

The above expression gives a simultaneous set of conditions for each basic variable for which one can solve those conditions. Two cases which arise across the set of conditions depending on the sign of individual elements in $B^{-1}r$.

$$\theta \geq -\frac{\left(B^{-1}b_{old}\right)_i}{\left(B^{-1}r\right)_i}, \quad \text{where } \left(B^{-1}r\right)_i > 0$$

and

$$\theta \leq -\frac{\left(B^{-1}b_{old}\right)_i}{\left(B^{-1}r\right)_i}, \quad \text{where } \left(B^{-1}r\right)_i < 0$$

much as in the row minimum rule where positive values of $B^{-1}r$ limit how negative θ can be and negative numbers limit how positive θ can become. This result shows the range over which θ can be altered with the basis remaining optimal.

Example

Suppose in the Joe's van factory example we wished to change the first right hand side.

Ordinarily, if one wishes to change the ith RHS, then r will be a vector with all zeros except for a one in the ith position, as illustrated below

$$r = \begin{bmatrix} 0 \\ . \\ . \\ . \\ 0 \\ 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix} \leftarrow i^{th} \text{element}$$

Thus when we change row 1 in our two row problem

$$b_{new} = \begin{bmatrix} 12 \\ 280 \end{bmatrix} + \theta \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 12 + \theta \\ 280 \end{bmatrix}$$

The resultant values of X_B becomes

$$X_{B_{new}} = \begin{bmatrix} 4 \\ 8 \end{bmatrix} + \theta \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ 8 \end{bmatrix} + \theta \begin{bmatrix} 5 \\ -4 \end{bmatrix} \geq 0$$

which implies

$$-.8 = \frac{-4}{5} \leq \theta \leq \frac{-8}{-4} = 2$$

Therefore the first right hand side can be changed up by 2 or down by 0.8 without the basis changing. Note that during this alteration the solution (B^{-1} b) does change, but B^{-1} does not. Furthermore, this gives a range of values for b_1 for which the marginal value of the resource ($C_B B^{-1}$) remains the same.

This approach also encompasses a generalization of the RHS ranging problem. Namely, suppose we wish to alter several RHS's at the same time. In this case, the change vector (r) does not have one entry but rather several. For example, suppose in Joe's van that Joe will add both capacity and an employee. In that case the change vector would look like the following:

$$b_{new} = \begin{bmatrix} 12 \\ 280 \end{bmatrix} + \theta \begin{bmatrix} 1 \\ 40 \end{bmatrix} = \begin{bmatrix} 12 & + & \theta \\ 280 & + & 40 \end{bmatrix}$$

3.3.3.2 Ranging Objective Function Coefficients

The analysis of ranging objective function coefficients is conceptually similar to RHS ranging. We seek to answer the question: what is the range of values for a particular objective function coefficient for which the current basis is optimal?

To examine this question, write the new objective function as the old objective function plus γ, which is a change magnitude, times T which is a direction of change vector.

$$C_{new} = C_{old} + \gamma T$$

Simultaneously, one has to write an expression for the objective function coefficients of the basic variables

$$C_{new} = C_{old} + \gamma T_B$$

where T_B gives the way the C_B's are altered. Subsequently, one can reexpress the restriction that the reduced cost values must be nonnegative as

$$\left(C_B B^{-1} a_j - c_j \right)_{new} = C_{Bnew} B^{-1} a_j - c_{jnew} \geq 0$$

which reduces to

$$\left(C_B B^{-1} a_j - c_j\right)_{new} = \left(C_B B^{-1} a_j - c_j\right)_{old} + \gamma\left(T_B B^{-1} a_j - T_j\right) \geq 0$$

In turn, we discover for nonbasic variables

$$\gamma \leq C_B B^{-1} a_j - c_j$$

while for basic variables

$$\gamma \leq \frac{-\left(C_B B^{-1} a_j - c_j\right)_{old}}{\left(T_B B^{-1} a_j - T_j\right)}, where\left(T_B B^{-1} a_j - T_j\right) < 0$$

$$\gamma \geq \frac{-\left(C_B B^{-1} a_j - c_j\right)}{\left(T_B B^{-1} a_j - T_j\right)}, where\left(T_B B^{-1} a_j - T_j\right) > 0$$

Example

Suppose in our example problem we want to alter the objective function on X_{fancy} so it equals $2{,}000 + \gamma$. The setup then is

$$C_{new} = [2000 \quad 1700 \quad 0 \quad 0] + \gamma[1 \quad 0 \quad 0 \quad 0]$$

and

$$T_B = [0 \quad 1]$$

so $C_{B_{new}} B^{-1} A - C_{new}$ for the nonbasics equals

$$[500 \quad 60] + \gamma\left[[0 \quad 1]\begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - [0 \quad 0]\right] = [500 \quad 60] + \gamma[-4 \quad 1/5] \geq 0$$

which implies $-300 \leq \gamma \leq 125$ or that the basis is optimal for any objective function value for X_{fancy} between 2125 and 1700. This shows a range of prices for X_{fancy} for which its optimal level is constant.

3.3.3.3 Changes in the Technical Coefficient Matrix

The above analysis examined changes in the objective function coefficients and right hand sides. It is possible that the technical coefficients of several decision variables may be simultaneously varied. This can be done simply if all the variables are nonbasic. Here we examine incremental changes in the constraint matrix. For example, a farmer might purchase a new piece of equipment which alters the labor requirements over several crop enterprises which use that equipment. In this section, procedures which allow analysis of simultaneous incremental changes in the constraint matrix are presented.

Consider the linear programming problem

$$
\begin{aligned}
\text{Max} \quad & Z = CX \\
\text{s.t.} \quad & \overline{A}X = b \\
& X = 0
\end{aligned}
$$

where the matrix of the technical coefficients is to be altered as follows

$$\overline{A} = A + M$$

where $\overline{A}$, A, and M are assumed to be mxn matrices. Suppose the matrix M indicates a set simultaneous changes to be made in A and that the problem solution is nondegenerate, possessing an unique optimal solution. Then the expected change in the optimal value of the objective function given M is

$$Z_{new} - Z_{old} = -U^* M X^*$$

where X^* and U^* are the optimal decision variable values (B^{-1} b) and shadow prices ($C_B B^{-1}$) for the unaltered original problem.

The equation is an approximation which is exact when the basis does not change. See Freund(1985) for its derivation and further discussion. Intuitively the equation can be explained as follows: since M gives the per unit change in the resource use by the variables, then MX^* gives the change in the resources used and $U^* M X^*$ then gives an approximation of the value of this change. Further, if M is positive, then more resources are used and the Z value should go down so a minus is used. McCarl, et al.(1990) investigated the predictive power of this equation and conclude it is a good approximation for the case they examined.

Illustrative Example

To illustrate the procedure outlined in the preceding section, consider the Joe's van shop model and suppose we wish to consider the effect of an equal change in the labor coefficients. For a change equal θ, the problem becomes

$$
\begin{aligned}
\textit{Max } Z(\theta)= \quad & 2000X_{fancy} & + & \quad 1700X_{fine} \\
\textit{s.t.} \quad & X_{fancy} & + & \quad X_{fine} & + & \quad S_1 & = 12 \\
& (25+\theta)X_{fancy} & + & \quad (20+\theta)X_{fine} & & + & S_2 = 12 \\
& X_{fancy}, \quad 3-19 & & \quad X_{fine}, & & S_1, & S_2 \geq 0
\end{aligned}
$$

For no change ($\theta = 0$), the optimal solution to this problem is

$$X^* = B^{-1}b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$$

$$U^* = C_B B^{-1} = \begin{bmatrix} 500 & 60 \end{bmatrix}$$

with the optimal value of the objective function equal to 22,800, our change matrix in this case is

$$M = \begin{bmatrix} 0 & 0 \\ \theta & \theta \end{bmatrix}.$$

Thus, the change in the value of the objective function is given by

$$Z_{new} - Z_{old} = -U^* M X^*$$

$$= -\begin{bmatrix} -500 & 60 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \theta & \theta \end{bmatrix} \begin{bmatrix} 4 \\ 8 \end{bmatrix} = -720\theta$$

Suppose the labor requirement is reduced by 1 hour for both vans so that $\theta = -1$, then the anticipated increase in the objective function that would result from using the new machine is

$$\Delta Z = -U^* M X^* = 720$$

Solution of the revised problem shows the objective function changes by 720.

3.3.4 Finding the Solution

As shown above, the linear programming solution contains a lot of information relative to the ways the objective function and basic variables change given changes in parameters. However, not all this information is included in an optimal solution as reported by modeling systems such as GAMS. Consider the following problem

$$\begin{array}{llllll}
Max\ Z = & 3x_1 & + & 2x_2 & + & 0.5x_3 & & (ZROW) \\
& x_1 & + & x_2 & + & x_3 & \leq 10 & (CONSTRAIN\ 1) \\
& x_1 & - & x_2 & & & \leq 3 & (CONSTRAIN\ 2) \\
& x_1, & & x_2 & & & \geq 0 &
\end{array}$$

The GAMS solution information for this problem appears in Table 3.1. The optimal objective function value equals 26.5. Then GAMS gives information on the equations. For this problem, there are 3 equations as named in the parenthetical statements above. For each equation information is given on the lower limit (labeled LOWER), value of AX^* (labeled LEVEL), upper limit (labeled UPPER), and shadow price $C_B B^{-1}$ (labeled MARGINAL). The objective function row (ZROW) does not contain interesting information. The constraint equations show there is a) no lower bound on the first equation (there would be if it were μ) b) the left hand side equals 10

(AX*) and c) the right hand side is 10 (UPPER) while the shadow price is 2.5 (MARGINAL). Similar information is present for the second equation.

Turning to the variables, the solution table gives the variable name, lower bound (LOWER), optimal level (LEVEL), upper bound (UPPER) and reduced cost (MARGINAL). The solution shows $X_1 = 6.5$ and $X_2 = 3.5$ while the cost of forcing X_3 into the solution is estimated to be $2.00 per unit. We also see the objective function variable (Z) equals 26.5. The solution information also indicates if an unbounded or infeasible solution arises.

GAMS output does not provide access to the B^{-1} or $B^{-1}a_j$ matrices. This is a mixed blessing. A 1000 row model would have quite large B^{-1} and $B^{-1}a_j$ matrices, but there are cases where it would be nice to have some of the information. None of the GAMS solvers provide access to this data.

3.3.5 Alternative Optimal and Degenerate Cases

Linear programming does not always yield a unique primal solution or resource valuation. Non-unique solutions occur when the model solution exhibits degeneracy and/or an alternative optimal.

Degeneracy occurs when one or more of the basic variables are equal to zero. Degeneracy is a consequence of constraints which are redundant in terms of their coefficients for the basic variables. Mathematically, given a problem with M rows and N original variables, and M slacks, degeneracy occurs when there are more than N original variables plus slacks that equal zero with less than M of the original variables and slacks being non-zero.

Most of the discussion in LP texts regarding degeneracy involves whether or not degeneracy makes the problem harder to solve and most texts conclude it does not. Degeneracy also has important implications for resource valuation. Consider for example the following problem:

$$\begin{array}{rlll}
\text{Max} & 100X_1 & + & 75X_2 \\
& X_1 & & \leq & 50 \\
& & X_2 & \leq & 50 \\
& X_1 & + & X_2 & \leq & 100
\end{array}$$

The solution to this problem is degenerate because the third constraint is redundant to the first two. Upon application of the simplex algorithm, one finds in the second iteration that the variable X_2 can be entered in place of the slack variable from either the second or third rows. If X_2 is brought into the basis in place of the second slack, the shadow prices determined are $(u_1, u_2, u_3) = (100, 75, 0)$. If X_2 is brought into the basis in place of the third slack, the value of the shadow prices are $(u_1, u_2, u_3) = (25, 0, 75)$. These differ depending on whether the second or third slack variable is in the basis at a value of zero. Thus, the solution is degenerate, since a variable in the basis (one of the slacks) is equal to zero (given three constraints there would be

three non-zero variables in a non-degenerate solution). The alternative sets of resource values may cause difficulty in the solution interpretation process.   For example, under the first case, one would interpret the value of the resource in the second constraint as $75, whereas in the second case it would interpret nominally as $0. Here the shadow prices have a direction and magnitude as elaborated in McCarl (1977) (this has been shown numerous times, see Drynan or Gal, Kruse, and Zornig.). Note that decreasing the RHS of the first constraint from 100 to 99 would result in a change in the objective function of 100 as predicted by the first shadow price set, whereas increasing it from 100 to 101 would result in a $25 increase as predicted by the first shadow price set. Thus, both sets of shadow prices are valid. The degenerate solutions imply multiple sets of resource valuation information with any one set potentially misleading. Both McCarl (1977) and Paris discuss approaches which may be undertaken in such a case. The underlying problem is that some of the right hand side ranges are zero, thus the basis will change for any right hand side alterations in one direction.

Another possibility in the simplex algorithm is the case of alternate optimal. An alternative optimal occurs when at least one of the nonbasic variables has a zero reduced cost; i.e., $C_B B^{-1} a_j - c_j$ for some $j \backslash NB$ equal to zero. Thus, one could pivot, or bring that particular variable in the solution replacing one of the basic variables without changing the optimal objective function value. Alternative optimals also imply that the reduced cost of more than M variables in a problem with M constraints are equal to zero. Consider the following problem:

$$\text{Max} \quad 25X_1 \; + \; 50X_2$$
$$X_1 \; + \; 2X_2 \; \leq \; 100$$

In this problem the optimal solution may consist of either $X_1 = 100$ or $X_2 = 50$ with equal objective function values one or the other of these variables will have zero reduced cost at optimality. Alternative optimals may cause difficulty to the applied modeler as there is more than one answer which is optimal for the problem. Paris (1981, 1991); McCarl et al. (1977); McCarl and Nelson, and Burton et al., discuss this issue further.

3.3.6 Finding Shadow Prices for Bounds on Variables

Linear programming codes impose upper and lower bounds on individual variables in a special way. Many modelers do not understand where upper or lower bound related shadow prices appear. An example of a problem with upper and lower bounds is given below.

$$\text{Max} \quad 3X_1 \; - \; X_2$$
$$\text{s.t.} \quad X_1 \; + \; X_2 \; \leq \; 15$$
$$X_1 \qquad\qquad \leq \; 10$$
$$X_2 \; \geq \; 1$$

The second constraint imposes an upper bound on X_1, i.e., $X_1 \leq 10$, while the third constraint, $X_2 \geq 1$, is a lower bound on X_2. Most LP algorithms allow one to specify these particular restrictions as either constraints or bounds. Solutions from LP codes under both are shown in Table 3.2.

In the first solution the model has three constraints, but in the second solution the model has only one constraint with the individual constraints on X_1 and X_2 imposed as bounds. Note that in the first solution there are shadow prices associated with constraints two and three. However, this information does not appear in the equation section of the second solution table. A closer examination indicates that while X_1 and X_2 are non-zero in the optimal solution, they also have reduced costs. Variables having both a non-zero value and a non-zero reduced cost are seemingly not in accordance with the basic/nonbasic variable distinction. However, the bounds have been treated implicitly. Variables are transformed so that inside the algorithm they are replaced by differences from their bounds and thus a nonbasic zero value can indicate the variable equals its bound. Thus, in general, the shadow prices on the bounds are contained within the reduced cost section of the column solution. In the example above the reduced costs show the shadow price on the lower bound of X_2 is 1 and the shadow price on the upper bound of X_1 is -3. Notice these are equal to the negative of the shadow prices from the solution when the bounds are treated as constraints.

One basic advantage of considering the upper and lower limits on variables as bounds rather than constraints is the smaller number of rows which are required.

3.4 Further Details on the Simplex Method

The simplex method as presented above is rather idealistic avoiding a number of difficulties. Here we present additional details regarding the basis in use, finding an initial nonnegative basis and some comments on the real LP solution method.

3.4.1 Updating the Basis Inverse

In step 5 of the matrix simplex method the basis inverse needs to be changed to reflect the replacement of one column in the basis with another. This can be done interatively using the so-called product form of the inverse (Hadley(1962)). In using this procedure the revised basis inverse (B^{-1}) is the old basis inverse (B^{-1}) times an elementary pivot matrix (P), i.e.,

$$B^{-1}_{new} = PB^{-1}_{old}$$

This pivot matrix is formed by replacing the $I^{*\text{th}}$ (where one is pivoting in row I^*) column of an identity matrix with elements derived from the column associated with the entering variable. Namely, suppose the entering variable column updated by the current basis inverse has elements

$$a^*_\eta = B^{-1}a_\eta.$$

then the elements of the elementary pivot matrix are

$$P_{i^*i^*} = +1/a^*_{i\eta}$$

$$P_{ki^*} = a^*_{k\eta}/a^*_{i\eta}, \ k \neq i^*$$

Suppose we update the inverse in the Joe's van example problem using product form of the inverse. In the first pivot, after X_{fancy} has been identified to enter the problem in row 2, then we replace the second column in an identity matrix with a column with one over the pivot element (the element in the second row of $B^{-1}a_\eta$ divided by the pivot element elsewhere. Since $B^{-1}a_\eta$

equals $\begin{vmatrix} 1 \\ 25 \end{vmatrix}$, the pivot matrix P_1 is

$$P_1 = \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix}$$

and the new B^{-1} is

$$B_1^{-1} = P_1 B_0^{-1} = \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 1/(1/5) & 0 \\ -(4/5)/(1/5) & 1 \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ -4 & 1 \end{bmatrix}$$

Similarly in the second pivot we find the minimum in the first row and have $B^{-1}a_\eta = \begin{bmatrix} 1/5 \\ 4/5 \end{bmatrix}$, so

that in forming P_2, the first column of an identity matrix was replaced since X_5 will enter as the first element of the basis vector. Multiplication of B^{-1} by P_2 gives

$$B_2^{-1} = P_1 B_0^{-1} = \begin{bmatrix} 5 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1/25 \\ 0 & 1/25 \end{bmatrix} = \begin{bmatrix} 5 & -1/5 \\ -4 & 1/5 \end{bmatrix}$$

which equals the basis inverse computed above.

3.4.2 Advanced Bases

The process of solving a LP is a hunt for the optimal basis matrix. Experience with LP reveals that the simplex method usually requires two or three times as many iterations as the number of constraints to find an optimal basis. This implies that when solving a series of related problems (i.e., changing a price of an input), it may be worthwhile to try to save the basis from one problem and begin the next problem from that particular basis. This is commonly supported in

LP solution algorithms and is quite important in applied LP involving sizable models. In a recent study, it took more than thirty-five hours of computer time to obtain an initial basis, but from an advanced basis, a series of related problems with a few changes in parameters could be solved in two hours. Dillon (1970) discusses ways of suggesting a basis for problems that have not previously been solved.

Modeling systems like GAMS do not readily take an advanced basis although one can be attempted by a choice of initial levels for variables (EXECUTE_LOADPOINT) permits this). However, once an initial model solution has been found, then any additional solutions are computed from an initial basis. Furthermore, an advanced basis can be employed by restarting from a stored file.

3.4.3 Finding an Initial Feasible Basis

When an LP problem includes only less-than constraints with non-negative right hand sides, it is straightforward to obtain an initial feasible basis with all non-negative variable values. In that case the slacks form the initial basis and all decision variables are nonbasic, equaling zero, with each slack variable set equal to the RHS ($s_i = b_i$). The initial basis matrix is an identity matrix. In turn, the simplex algorithm is initiated.

However, if one or more: a) negative right hand sides, b) equality constraints, and/or c) greater than or equal to constraints are included, it is typically more difficult to identify an initial feasible basis. Two procedures have evolved to deal with this situation: the Big M method and the Phase I/Phase II method. Conceptually, these two procedures are similar, both imply an inclusion of new, <u>artificial variables</u>, which artificially enlarge the feasible region so an initial feasible basis is present. The mechanics of artificial variables, of the Big M method and the Phase I/ Phase II problem are presented in this section.

Models which contain negative right hand sides, equality and or greater than constraints do not yield an initial feasible solution when all X's are set to zero. Suppose we have the following

$$
\begin{array}{rlcl}
\text{Max} & \text{CX} & & \\
\text{s.t.} & \text{RX} & \leq & \text{b} \\
& \text{DX} & \leq & \text{-e} \\
& \text{FX} & = & \text{g} \\
& \text{HX} & \geq & \text{p} \\
& \text{X} & \geq & 0
\end{array}
$$

general problem where b, e, and p are positive.

Conversion of this problem to the equality form requires the addition of slack, surplus and artificial variables. The slack variables (S_1 and S_2) are added to the first and second rows (note that while we cover this topic here, most solvers do this automatically). <u>Surplus variables</u> are needed in the last constraint type and give the amount that left hand side (HX) exceeds the right hand side limit (p). Thus, the surplus variables (W) equal p - HX and the constraint becomes HX - W = p.

The resultant equality form becomes

$$
\begin{array}{llllllll}
\text{Max} & CX & + & O_1S_1 & + & O_2S_2 & + & O_4W \\
& RX & + & I_1S_1 & & & & & = & b \\
& DX & & & + & I_2S_2 & & & = & -e \\
& FX & & & & & & & = & g \\
& HX & & & & & & I_4W & = & p \\
& X, & & S_1, & & S_2, & & W & \geq & 0
\end{array}
$$

Where the I's are appropriately sized identity matrices and the O's are appropriately sized vectors of zeros.

Note that when X = 0, the slacks and surplus variables do not constitute an initial feasible basis. Namely, if S_2 and W are put in the basis; then assuming e and p are positive, the initial solution for these variables are negative violating the non-negativity constraint

S_2 = -e and W = -p.

Furthermore, there is no apparent initial basis to specify for the third set of constraints (FX = g). This situation requires the use of <u>artificial variables</u>. These are variables entered into the problem which permit an initial feasible basis, but need to be removed from the solution before the solution is finalized.

Artificial variables are entered into each constraint which is not satisfied when X=0 and does not have an easily identified basic variable. In this example, three sets of artificial variables are required.

$$
\begin{array}{lllllllllllll}
\text{Max} & CX & + & O_1S_1 & + & O_2S_2 & + & O_4W \\
& RX & + & I_1S_1 & & & & & & & & & = & b \\
& DX & & & + & I_2S_2 & & & - & I_2A_2 & & & = & -e \\
& FX & & & & & & & + & I_3A_3 & & & = & g \\
& HX & & & & & I_4W & & & & + & I_4A_4 & = & p \\
& X, & & S_1, & & S_2, & & W, & & A_2, & & A_3, & & A_4 & \geq & 0
\end{array}
$$

Here A_2, A_3, and A_4 are the artificial variables which permit an initial feasible nonnegative basis but which must be removed before a "true feasible solution" is present. Note that S_1, A_2, A_3, and A_4 can be put into the initial basis. However, if elements of A_3 are nonzero in the final solution, then the original $FX = g$ constraints are not satisfied. Similar observations are appropriate for A_2 and A_4. Consequently, the formulation is not yet complete. The objective function must be manipulated to cause the artificial variables to be removed from the solution. The two alternative approaches reported in the literature are the BIG M method and the Phase I/Phase II method.

3.4.3.1 BIG M Method

The BIG M method involves adding large penalty costs to the objective function for each artificial variable. Namely, the objective function of the above problem is written as

$$\text{Max } CX + O_1S_1 + O_2S_2 + O_4W - M_2A_2 - M_3A_3 - M_4A_4$$

where M_2, M_3, and M_4 are conformable sized vectors of large numbers that will cause the model to drive A_2, A_3, and A_4 out of the optimal solution.

An example of this procedure involves the problem

$$
\begin{array}{rcrcll}
\text{Max} & 3x_1 & + & 2x_2 & & \\
& x_1 & + & 2x_2 & \leq & 10 \\
+ & x_1 & + & x_2 & \leq & -2 \\
- & x_1 & + & x_2 & = & 3 \\
& x_1 & + & x_2 & \geq & 1 \\
& x_1, & & x_2 & \geq & 0
\end{array}
$$

and the model as prepared for the Big M method is in Table 3.3. The optimal solution to this problem is in Table 3.4.

This solution is feasible since A_2, A_3, and A_4 have been removed from the solution. On the other hand, if the right hand side on the second constraint is changed to -4, then A_2 cannot be forced from the solution and the problem is infeasible. This, with the BIG M method one should note that the artificial variables must be driven from the solution for the problem to be feasible so M must be set large enough to insure this happens (if possible).

3.4.3.2 Phase I/Phase II Method

The Phase I/Phase II method is implemented in almost all computer codes. The procedure involves the solution of two problems. First, (Phase I) the problem is solved with the objective function replaced with an alternative objective function which minimizes the sum of the artificial variables, i.e.,

$$\text{Min } L_2A_2 + L_3A_3 + L_4A_4$$

where L_i are conformably sized row vectors of ones.

If the Phase I problem has a nonzero objective function (i.e., not all of the artificials are zero when their sum has been minimized), then the problem does not have a feasible solution. Note this means the reduced cost information in an infeasible problem can correspond to this modified objective function. Otherwise, drop the artificial variables from the problem and return to solve the real problem (Phase II) using the Phase I optimal basis as a starting basis and solve using the normal simplex procedure.

The addition of the slack, surplus and artificial variables is performed automatically in almost all solvers including all that are associated with GAMS.

3.4.4 The Real LP Solution Method

The above material does not fully describe how a LP solution algorithm works. However, the algorithm implemented in modern computer codes, while conceptually similar to that above is operationally quite different. Today some codes use interior point algorithms combined with the simplex method (for instance, OSL, Singhal et al.). Codes also deal with many other things such as compact data storage, basis reinversion, efficient pricing, and round-off error control (e.g., see Orchard-Hays or Murtagh).

In terms of data storage, algorithms do not store the LP matrix as a complete MXN matrix. Rather, they exploit the fact that LP problems often be sparse, having a small number of non-zero coefficients relative to the total possible number, by only storing non-zero coefficients along with their column and row addresses. Further, some codes exploit packing of multiple addresses into a single word and economize on the storage of repeated numerical values (for in-depth discussion of data storage topics see Orchard-Hays or Murtagh).

Perhaps the most complex part of most modern day LP solvers involves inversion. As indicated above, the B^{-1} associated with the optimal solution is needed, but in forming B^{-1} the code usually performs more iterations than the number of constraints. Thus, the codes periodically construct the basis inverse from the original data. This is done using product form of the inverse; but this also involves such diverse topics as LU decomposition, reduction of a matrix into lower triangular form and matrix factorization. For discussion in these topics see Murtagh.

LP codes often call the formation of reduced costs the pricing pass and a number of different approaches have been developed for more efficient computation of pricing (see Murtagh for discussion).

Finally, LP codes try to avoid numerical error. In computational LP, one worries about whether numbers are really non-zero or whether rounding error has caused fractions to compound giving false non-zeros. Solver implementations usually make extensive use of tolerances and basis reinversion schemes to control such errors. Murtagh and Orchard-Hays discuss these.

The purpose of the above discussion is not to communicate the intricacies of modern LP solvers, but rather to indicate that they are far more complicated than the standard implementation of the simplex algorithm as presented in the first part of the chapter.

References

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

Burton, R.O., J.S. Gidley, B.S. Baker and K.J. Reda-Wilson. "Nearly Optimal Linear Programming Solutions: Some Conceptual Issues and a Farm Management Application." American Journal of Agricultural Economics. 69(1987):813-818.

Dantzig, G.B. Linear Programming and Extension. Princeton, New Jersey: Princeton University Press, 1963.

Dillon, M. "Heuristic Selection of Advanced Bases for a Class of Linear Programming Models." Operation Research. 18(1970):90-100.

Drynan, R.G. "Forum on Resolving Multiple Optima in Linear Programming." Review of Marketing and Agricultural Economics. 54(1986):31-35.

Freund, R.M. "Postoptimal Analysis of a Linear Program Under Simultaneous Changes in Matrix Coefficients." Mathematical Programming Studies. 24(1985):1-13.

Gal, T., H.J. Kruse, and P. Zornig. "Survey of Solved and Open Problems in the Degeneracy

Phenomenon." Mathematical Programming. 42(1988):125-133.

Hadley, G. Linear Programming. Reading, Mass.: Addison-Wesley Publishing Co., 1962.

Karmarkar, N. "A New Polynomial-Time Algorithm for Linear Programming." Combinatoria. 4(1984):373-395.

McCarl, B.A. "Degeneracy, Duality and Shadow Prices in Linear Programming." Canadian Journal of Agricultural Economics. 25(1977):70-73.

McCarl, B.A., W.V. Candler, D.H. Doster, and P. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning." Canadian Journal of Agricultural Economics. 25(1977):17-30.

McCarl, B.A., D.E. Kline, and D.A. Bender. "Improving on Shadow Price Information for Identifying Critical Farm Machinery." American Journal of Agricultural Economics. 72(1990):582-588.

McCarl, B.A. and C.H. Nelson. "Multiple Optimal Solutions in Linear Programming Models: Comment." American Journal of Agricultural Economics. 65(1983):181-83.

Murtagh, B.A. Advanced Linear Programming. McGraw-Hill Book Co., New York, 1981.

Orchard-Hays, W. Advanced Linear- Programming Computing Techniques. McGraw-Hill, Inc: New York, 1968.

Optimization Subroutine Library (OSL). IBM Guide and Reference, Release 2. Ed. J.A. George and J.W.H. Liu. IBM Corporation, 1990-91.

Paris, Q. "Multiple Optimal Solutions in Linear Programming Models." American Journal of Agricultural Economics. 63(1981):724-27.

Paris, Q. An Economic Interpretation of Linear Programming. Iowa State University Press: Ames, Iowa, 1991.

Singhal, J.A., R.E. Marsten, and T.L. Morin. "Fixed Order Branch-and-Bound Methods for Mixed-Integer Programming: The ZOOM System." Department of Management Information Systems, University of Arizona, Tucson, AZ, 85721, December, 1987

Table 3.1. GAMS Solution of Example Model

```
SOLVE SUMMARY

MODEL      PROBLEM   OBJECTIVE Z
TYPE       LP        DIRECTION MAXIMIZE
SOLVER   MINOS5      FROM LINE  37

**** SOLVER STATUS  1 NORMAL COMPLETION
**** MODEL STATUS   1 OPTIMAL
**** OBJECTIVE VALUE     26.5000

 EXIT -- OPTIMAL SOLUTION FOUND

                          LOWER      LEVEL      UPPER     MARGINAL
---- EQU ZROW          .          .          .          1.000
---- EQU CONSTRAIN1  -INF      10.000     10.000        2.500
---- EQU CONSTRAIN2  -INF       3.000      3.000        0.500

ZROW             OBJECTIVE FUNCTION
CONSTRAIN1       FIRST CONSTRAINT
CONSTRAIN2       SECOND CONSTRAINT

            LOWER      LEVEL     UPPER     MARGINAL

---- VAR X1     .       6.500     +INF         .
---- VAR X2     .       3.500     +INF         .
---- VAR X3     .         .       +INF     -2.000
---- VAR Z   -INF     26.500     +INF         .

X1    FIRST VARIABLE
X2    SECOND VARIABLE
X3    THIRD VARIABLE
Z     OBJECTIVE FUNCTION
```

Table 3.2.   Solution with Bounds Imposed as Constraints and as Bounds

Solved with Constraints

Variable	Value	Reduced Cost	Status	Equation	Level	Shadow Price	Status
X_1	10	0	Basic	1	4	0	Basic
X_2	1	0	Basic	2	0	3	
				3	0	-1	

Solved with Bounds

Variable	Value	Reduced Cost	Status	Equation	Level	Shadow Price	Status
X_1	10	-3	(U)	1	4	0	Basic
X_2	1	1	(L)				

Table 3.3.   The Model as Ready for the Big M Method

Max	$3x_1$	+	$2x_2$	+	$0S_1$	+	$0S_2$	+	$0W$	-	$99A_2$	-	$99A_3$	-	$99a_4$		
	x_1	+	x_2	+	S_1											=	10
	x_1	-	x_2			+	S_2			-	A_2					=	-2
	$-x_1$	+	x_2									+	A_3			=	3
	x_1	+	x_2					-	W					+	A_4	=	1
	x_1	,	x_2	,	S_1	,	S_2	,	W	,	A_2	,	A_3	,	A_4	$\geq$	0

Table 3.4.   Solution to the Big M Problem

Variable	Value	Reduced Cost	Equation	Shadow Price
x_1	1.333	0	1	1.667
x_2	4.333	0	2	0
S_1	0	1.667	3	-1.333
S_2	1	0	4	0
W	4.667			
A_2	0	-99		
A_3	0	-97.667		
A_4	0	-99		