DIVISION OF AGRICULTURAL SCIENCES
AGRICULTURAL EXPERIMENT STATION
UNIVERSITY OF CALIFORNIA

Calculating Frontier Multi-Product, Multi-Factor Production and Cost Relationships—A Computerized Algorithm—

by DARYL CARLSON

GIANNINI FOUNDATION OF AGRICULTURAL ECONOMICS

October 1976

University of California, Davis
Department of Agricultural Economics

CALCULATING FRONTIER MULTI-PRODUCT, MULTI-FACTOR PRODUCTION AND COST RELATIONSHIPS - A COMPUTERIZED ALGORITHM -

by

Daryl Carlson

May 1976

TABLE OF CONTENTS

Page

Introduction 1

Concepts of Economic Efficiency 2

A Graphic Approach 8

The "Farrell" Approach 11

Least-Cost Modification 19

The Linear Programming Computer Program 21

Modes of Operation 22

Control Parameters 25

Input Deck Structure 26

Illustrative Uses of the Program 28

Example 1: A general efficiency run 29

Example 2: A specific efficiency run 32

Example 3: A production surface run 35

Example 4: A production surface run with variable means 37

Example 5: A total economic efficiency run 40

Example 6: A cost surface run 43

The Computer Program 45

References 58

Calculating Efficient Multi-Product, Multi-Factor Production and Cost Relationships - A Computerized Algorithm -

When estimating relationships between inputs and outputs of production processes, it is often desirable to estimate the efficient or frontier relationship rather than the average. Least-squares regression methods are usually used for estimating average production and cost functions. Extensive literature exists on functional form problems, multiple product specification difficulties, and multicollinearity problems associated with standard regression approaches. Much less attention has been given to the problem of estimating efficient production and cost relationships although, from a theoretical point of view, the efficient production function is of considerable interest.

Two methods for calculating efficient production and cost relationships have been developed. Constrained-residuals regression was originally suggested by D. Aigner and S. Chu (1968) and has been implemented by C. Timmer (1971). This method consists of constraining all of the regression residuals to have the same sign and as a result the estimated function is forced to the "frontier" of the observations. For production function estimation with the residuals equal to the predicted output minus the actual output, the residuals would be constrained to be nonnegative. The term "frontier" will be used in this paper rather than "efficient" to denote those firms that use the minimum levels of inputs for given levels of outputs and for other given firm characteristics. Since the frontier relationships are only efficient relative to the observed firms, the underlying, truly efficient relationships cannot be determined from cross-sectional data.

The other production frontier computational approach was originated by M. Farrell (1957, 1962), extended by J. Boles (1967, 1972), and applied by W. Seitz (1971), B. Sitorus (1966), and D. Carlson (1972, 1975). Essentially, Farrell's method is to plot the observations (firms) as points in a space of as many dimensions as there are variables included in the analysis, to form the convex hull of this set of points, and to take the appropriate part of the surface of the convex hull as the estimate of the frontier relationship between all of the variables. The work by Farrell, Boles, and Seitz has concentrated on the development of the method to compute efficiency indices for each observed firm within a given sample.

The purpose of this paper is to further describe and extend Farrell's technique following the linear programming approach developed by J. Boles, to describe the procedures for using a computer program that calculates efficiency indices and frontier production and cost relationships, and to illustrate the use of the program with several examples. The computerized algorithm described in this paper differs from the algorithm developed by J. Boles (1971) in one important dimension. The procedure has been generalized to handle several "qualitative" factors in addition to the inputs and outputs of the production process. This capability is extremely useful for exploratory work with poorly defined production processes. Also the computer program described in the paper is much simpler to operate for an individual unfamiliar with linear programming.

Concepts of Economic Efficiency:

Defining measures of efficiency is an extremely difficult task for production processes involving more than one input and one output.

The problem is further complicated when scale and output quality is included into the specification of the production process. Following the approach by S. Danø (1966) and J. Henderson and R. Quandt (1971), the general implicit production function is written as

$$F(X_1, \ldots, X_I; Y_1, \ldots, Y_J; Q_1, \ldots, Q_K; S) = 0$$

where: X_i is input i

Y_j is output j

Q_k is quality factor k

S is a measure of the scale of the production process.

For the case where there are I inputs, one output, no quality factors and constant returns to scale the production function, $Y_1 = f(X_1, \ldots, X_I)$, is defined as the locus of the maximum output levels for alternative combinations of inputs. From the definition of the production function in this simple case, the measure of technical efficiency for firm n is given by:

$$TE_n = \frac{Y_{1n}}{f(X_{1n}, \ldots, X_{In})} .$$

Similarly, if input prices, C_i, are specified, the total economic efficiency of firm n can be measured by:

$$EE_n = \frac{\underset{f(X_1, \ldots, X_I) = Y_{1n}}{\underset{X_1, \ldots, X_I}{\min}} \left[\sum_{i=1}^{I} \frac{X_i}{Y_{1n}} . C_i \right]}{\sum_{i=1}^{I} \frac{X_{in}}{Y_{1n}} . C_i} ,$$

This definition of economic efficiency is based on the assumption that all firms face the same input prices. This definition can be generalized by replacing C_i in the above equation with C_{in}. However if input prices are different across firms, the resulting measure of economic efficiency indicates something other than production efficiency. Such a measure combines the efficiency of the firm's ability to obtain inexpensive inputs with production efficiency.

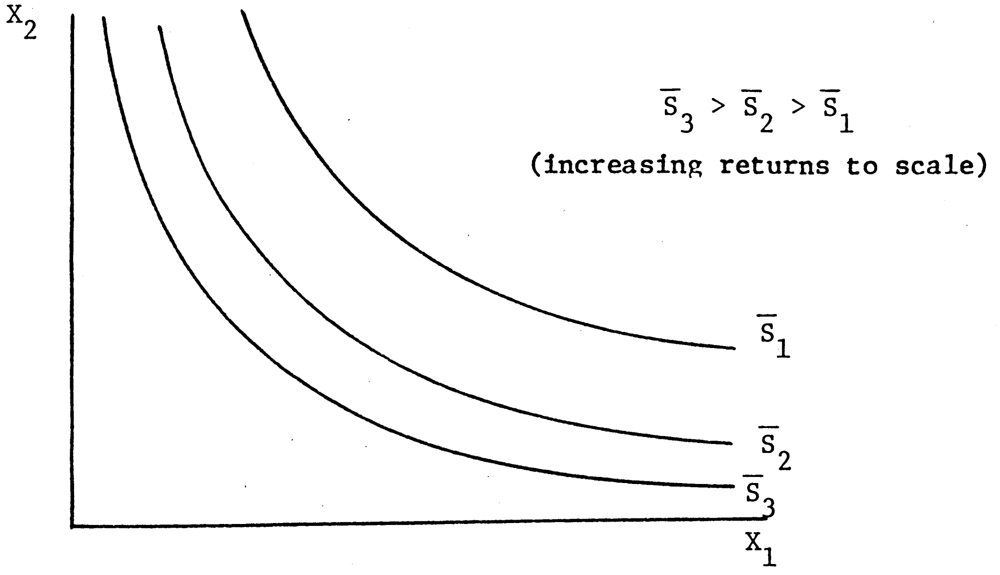
Since total economic efficiency (EE) is equal to the product of the technical efficiency measure (TE) and a measure of price or allocative efficiency (AE), the latter efficiency measure can be calculated for firm n as follows:

$$AE_n = EE_n/TE_n .$$

The above three definitions provide measures of the standard types of efficiency for a production process with I inputs, one output, and constant returns to scale. These measures assume that the production function, $Y_1 = f(X_1, \ldots, X_I)$, is known or can be estimated.

If the production process is extended to allow for nonconstant returns to scale, the production function, $Y_1 = f(X_1, \ldots, X_I; S)$, is defined as the locus of the maximum output levels for alternative combinations of inputs and for different scales of operation (Henderson and Quandt, 1971). In other words, there are a family of production functions; one for each size of firm. To illustrate, Figure 1 shows an isoquant for two of the inputs and several levels of scale (S).

Figure 1: Isoquants for Alternative Scales of Operation


X_2
$\overline{S}_3 > \overline{S}_2 > \overline{S}_1$
(increasing returns to scale)
$\overline{S}_1$
$\overline{S}_2$
$\overline{S}_3$
X_1

For this production situation, the measure of short-run technical efficiency for firm n, TE_n^s, follows directly from the constant returns case:

$$TE_n^s = \frac{Y_{1n}}{f(X_{1n}, \ldots, X_{In}; S_n)} \ .$$

And similarly for the measure of short-run economic efficiency, EE_n^s:

$$EE_n^s = \frac{\underset{\substack{X_1, \ldots, X_I \\ f(X_1, \ldots, X_I; S_n) = Y_{1n}}}{\min} \left[\sum_{i=1}^{I} \frac{X_i}{Y_{1n}} . C_i \right]}{\sum_{i=1}^{I} \frac{X_{in}}{Y_{1n}} . C_i} \ .$$

These definitions of short-run efficiency are based on the assumption that in the short-run the firm can not change its scale of operation. For the case of nonconstant returns to scale, the long-run measures of technical and economic efficiency are the same as the measures for the constant returns to scale situation. In the long run it is possible for the firm to adjust its scale of operation and therefore its efficiency should be measured relative to the optimal scale of operation.

Following the approach of S. Danø (1966), quality parameters can be included into the production function in the same manner as Henderson and Quandt include the scale parameter. As illustrated in Figure 1, a firm's efficiency should be determined relative to the production function for that firm's scale of operation. Similarly, a firm's efficiency should be determined relative to a production function with the same qualitative characteristics as that of the particular firm. The measures of short-run technical and economic efficiency are given by:

$$TE_n^s = \frac{Y_{1n}}{f(X_{1n}, \ldots, X_{In}; Q_{1n}, \ldots, Q_{Kn}; S_n)}$$

$$EE_n^s = \frac{\underset{X_1, \ldots, X_n}{\min} f(X_1, \ldots X_I; Q_{1n}, \ldots, Q_{kn}; S_n) - Y_{1n} \left[\sum_{i=1}^{I} \frac{X_i}{Y_{1n}} . C_i \right]}{\sum_{i=1}^{I} \frac{X_{in}}{Y_{1n}} . C_i} .$$

The above definitions and measures of efficiency span all cases except those production processes with several outputs and several inputs. Ignoring for a moment nonconstant returns to scale and quality parameters, it is still difficult to define general measures of technical and economic efficiency for a multiple output, multiple input production process. Several approaches have been suggested but none are very appropriate for many applications. One procedure is to construct an output index, Y*, by applying weights to each of the individual outputs. The usual approach is to use output prices to generate a total revenue variable to be used as an index. This procedure assumes that all firms face the same output prices and makes it impossible to separate technical efficiency from economic efficiency with respect to the mix of outputs produced. That is, if a firm is shown to be inefficient with this measure, it might be the case that the firm is efficiently producing its set of outputs but that the firm is producing the wrong mix of outputs given the prices of the outputs.

A second procedure for dealing with the multiple output, multiple input case is to construct an input index, X*, by applying weights to each of the individual inputs. This approach is the input complement to the above procedure and it suffers from the same problem.

A third technique is to base the efficiency measure on one of the outputs with all the other outputs fixed at specified levels. This approach is useful in situations where one particular output is of primary interest. The resulting measure of technical efficiency with respect to an output r for firm n, TE_n^r, is given by:

$$TE_n^r = \frac{Y_{rn}}{f(X_{1n}, \ldots, X_{In}; Y_{1n}, \ldots, Y_{r-1,n}, Y_{r+1,n}, \ldots, Y_J)} .$$

Similarly, the measure of economic efficiency is:

$$EE_n^r = \frac{\min\limits_{\substack{X_1, \ldots, X_I \\ Y_1, \ldots, Y_{r-1}, Y_{r+1}, \ldots, Y_J \\ f(X_1, \ldots, X_I; Y_1, \ldots, Y_{r-1}, Y_{r+1}, \ldots, Y_J) = Y_{rn}}} \left[\sum_{i=1}^{I} \frac{X_i}{Y_{rn}} .C_i \right]}{\sum_{i=1}^{I} \frac{X_{in}}{Y_{rn}} .C_i}$$

The difficulty with the above efficiency measures is that each firm will quite likely have different relative measures of efficiency depending upon the output chosen as the basis.

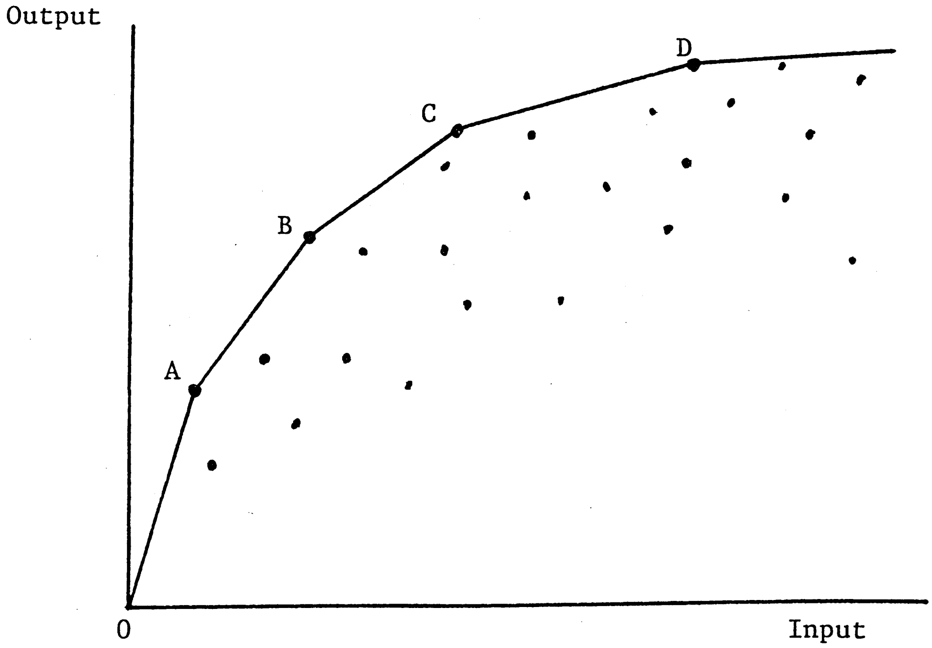
A fourth approach to this multiple input, output problem is one of decomposition. If the production process under study is not truly a joint production process, then it may be possible to separate the problem into an analysis of each output relative to the inputs used for that output alone. In this manner, the problem reduces into a form that can be handled with the single output measures discussed earlier. A firm's efficiency would then be determined for the production of each output separately.

The most general and satisfactory method for computing efficiency measures for joint, multiple input-output processes has been developed by M. Farrell (1957) but has been surprisingly neglected in empirical applications. His measures of efficiency completely generalize to the multiple input, multiple output, nonconstant returns to scale production process with quality dimensions. The next section presents a graphical description of Farrell's approach and the fourth section describes the computational algorithm required to implement his technique.

A Graphic Approach

Since Farrell's approach is not based on a statistically estimated equation but rather operates directly on the basic data, it is helpful to describe the technique from a graphical perspective. If the production process involves one input and one output, the production function can be drawn as shown in Figure 2.

Figure 2: The Farrell Production Function



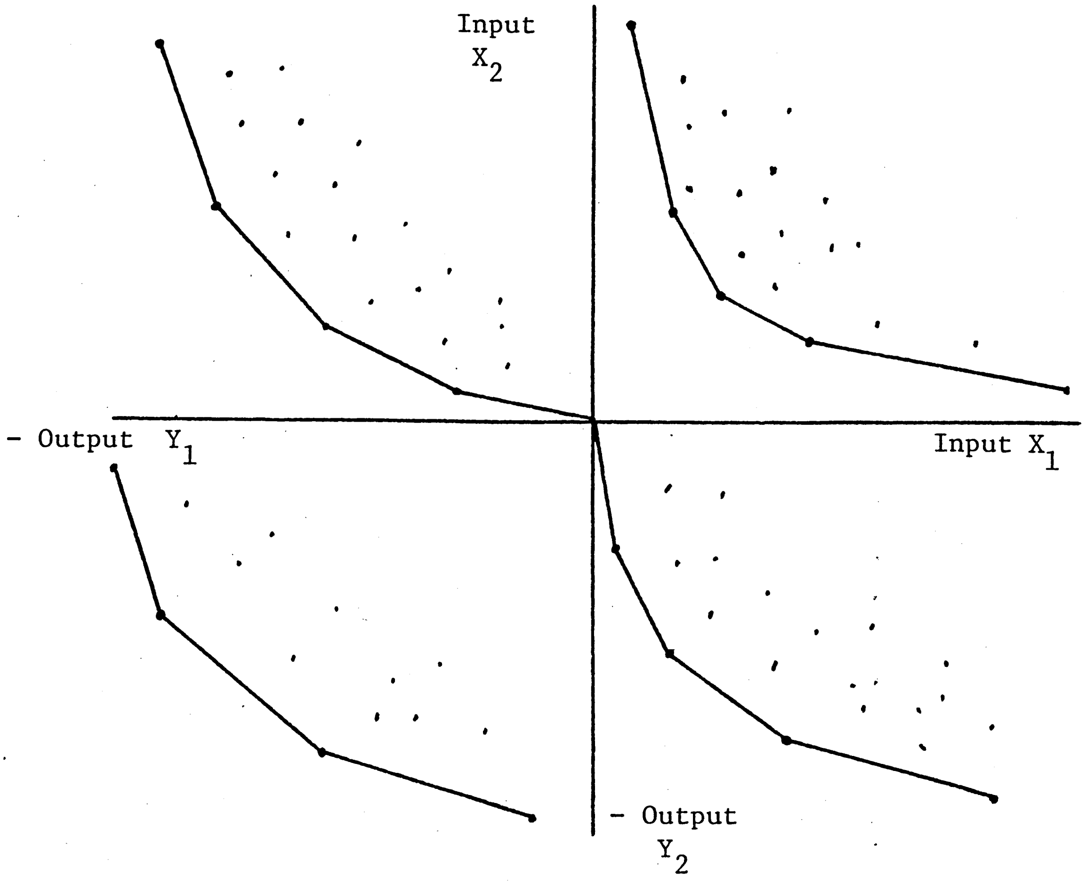
Each plotted point represents a firm and the production function, as determined by Farrell's method, is given by the curve OABCD. That is, the points on this curve represent the maximum output observed for a given level of input or, alternatively, the minimum amount of input observed for a given level of output. To define the curve OABCD as the production function, it is necessary to assume that the producton function is convex. This assumption implies that if two points are attainable in practice (for example, B and C), then so is any point representing a weighted average of them (points on the line connecting B and C). It must also be assumed that the production process is non-stochastic and that the variables are measured with no error. Calculation of the production function in this manner is obviously sensitive to the accuracy of the data. Since by definition of the production function, the desired relationship is to be at the extremes of the data, it is difficult to avoid this problem. Extreme caution must be taken with the data used in an analysis of this type.

This graphical approach of determining the production function has one very important advantage over statistical techniques. In order to estimate a production function with regression techniques, it is **necessary** that a functional form be specified. With the graphical method this requirement is not necessary as the data determines the shape of the relationships between all of the inputs and outputs.

The production function relationships between different inputs and alternative outputs may be desired, so a consistent method of constructing the production surface is needed. In order to accomplish this graphically as well as computationally, it is necessary to treat the input variables as positive and the output variables as negative. These relationships are illustrated in Figure 3.

Note that for all of the relationships, input versus input, input versus output, and output versus output, the desired production curve is the southwest portion of the outer ring circumscribing the scatter of points. The familiar isoquant relationship between two inputs (these inputs must be scaled by output to be true isoquants) of the production process appears in quadrant I of the graph in Figure 3. Productivity curves are shown in quadrants II and IV and the output transformation curve appears in the third quadrant. Since the outputs are specified as negative, the transformation curve as drawn in Figure 3 is inverted from the standard form.

Figure 3: The Multiple Input-Output Graphic Approach



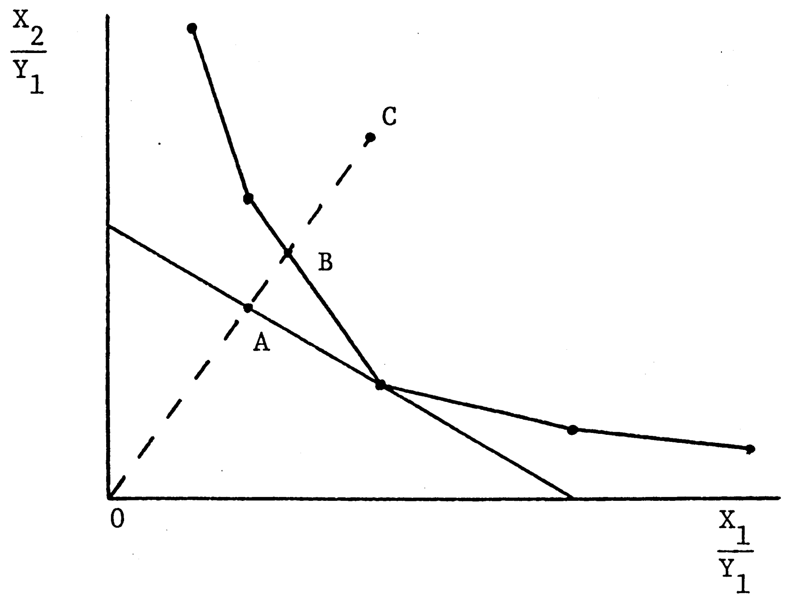
For a two input, one output production process, the measures of technical, economic, and allocative efficiency are drawn in Figure 4 and calculated as below:

$$TE = \frac{OB}{OC} \leq 1$$

$$EE = \frac{OA}{OC} \leq 1$$

$$AE = \frac{EE}{TE} = \frac{OA}{OC} \cdot \frac{OC}{OB} = \frac{OA}{OB} \leq 1 .$$

Figure 4: Graphical Efficiency Measures

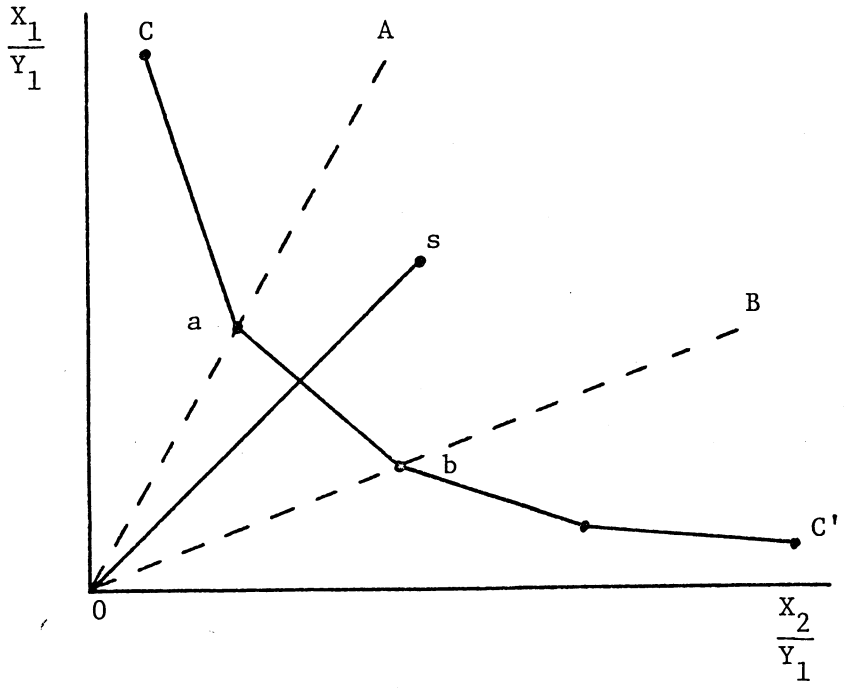


Although this graphical procedure can be completely generalized to several outputs and several inputs by expanding the number of dimensions of the graph, it is obviously not possible to draw the relationships. It is at this point that Farrell's computational method must be introduced. His procedure makes it possible to calculate portions (or slices) of the multi-dimensional production surface which can be graphed in two dimensions.

The "Farrell" Approach

For the case of many variables, the computational method by M. Farrell (1957, 1962) provides an efficient procedure for generating relationships like those illustrated in Figures 2-4. To determine the frontier relationships with Farrell's basic approach, J. Boles (1971, 1972) greatly simplified the computations required by formulating the procedure in terms of a linear programming problem. The link between the graphic approach illustrated above and the linear programming approach can best be made for the case of one output variable and two inputs. The desired relationship is illustrated in Figure 5.

Figure 5: Illustration of the Computational Approach



To interpret the input isoquant graphically, the two input variables should be scaled by the output variable. The relationship between the two input variables with all the other variables held constant is desired. To locate the observed firms that determine the frontier relationship between the two input variables, each scaled by the output variable, the procedure is to express the coordinates of each firm as a linear function of the coordinates of the other firms that lie closest to the origin of the graph in Figure 5. That is, find two firms (a and b) for each firm, s, in the sample such that:

$$z_a \frac{X_{1a}}{Y_{1a}} + z_b \frac{X_{1b}}{Y_{1b}} = \frac{X_{1s}}{Y_{1s}}$$

$$z_a \frac{X_{2a}}{Y_{1a}} + z_b \frac{X_{2b}}{Y_{1b}} = \frac{X_{2s}}{Y_{1s}}$$

and $(z_a + z_b)$ is a maximum over all possible pairs of firms a and b. The two firms that satisfy the above maximization problem lie on curve CC' in Figure 5. To force $(z_a + z_b)$ to the maximum, it is necessary that the two observations closest to the origin of Figure 5 be selected as firms a and b. It is also necessary that the two observations span firm s. That is, point s must lie between rays OA and OB in Figure 5. If firm s lies on the curve, the solution to the above problem is with $z_s = 1.0$ and the rest of the z's equal to zero. This solution follows since if firm s is on the frontier there will not exist any observations between the origin and point s or any weighted average of points between s and the origin unless there are identical observations or weighted averages of observations that are identical to s.

By defining the variables,

X_{it} = the quantity of the i^{th} input used by the t^{th} firm

Y_t = the quantity of output of the t^{th} firm

the above maximization problem for T firms can be written in a linear programming framework as:

$$\text{Maximize} \quad \sum_{t=1}^{T} z_t$$

$$\text{Subject to} \quad \sum_{t=1}^{T} z_t \frac{X_{it}}{Y_t} \leq \frac{X_{is}}{Y_s} \qquad i = 1,2$$

$$z_t \geq 0 \qquad t = 1, \ldots, T.$$

The procedure for determining those firms that are on the production surface CC' (as drawn in Figure 5) is to solve the above LP problem once for each firm in the analysis. Each time that the LP is solved a different firm is placed on the right-hand side of the constraints. All firms (including the one on the right-hand side) are included on the left-hand side of the constraints. If the solution with a particular firm on the right-hand side specifies a z value of one for that firm and a z value of zero for all of the other firms, that firm is on the production surface. In this manner, all of the firms that lie on the production surface can be identified. By definition, these firms are the technically efficient firms.

This simple, three-variable model can be generalized to include several input variables, other output variables, quality dimensions, and a scale parameter. These latter variables allow other aspects of the production process (besides inputs and outputs) to be included into the production or cost function specification. Let

Y_{rt} = the quantity of output r of the t^{th} firm

Q_{kt} = the k^{th} quality factor for the t^{th} firm

S_t = the scale parameter for the t^{th} firm,

then the general linear programming model for T firms, I input variables, J output variables, and K quality variables is written as:

$$\text{Maximize} \quad \sum_{t=1}^{T} z_t$$

Subject to:

[1] Input constraints

$$\sum_{t=1}^{T} z_t X_{it} \leq X_{is} \qquad i = 1, \ldots, I$$

[2] Output constraints

$$\sum_{t=1}^{T} z_t Y_{jt} \geq Y_{js} \qquad j = 1, \ldots, J$$

[3] Quality constraints

$$\frac{\sum_{t=1}^{T} z_t Q_{kt}}{\sum_{t=1}^{T} z_t} \geq Q_{ks} \qquad k = 1, \ldots, K$$

[4] Scale constraint

$$\frac{\sum_{t=1}^{T} z_t S_t}{\sum_{t=1}^{T} z_t} \left[\begin{matrix} \leq \\ = \\ \geq \end{matrix}\right] S_s$$

[5] Nonnegative constraint

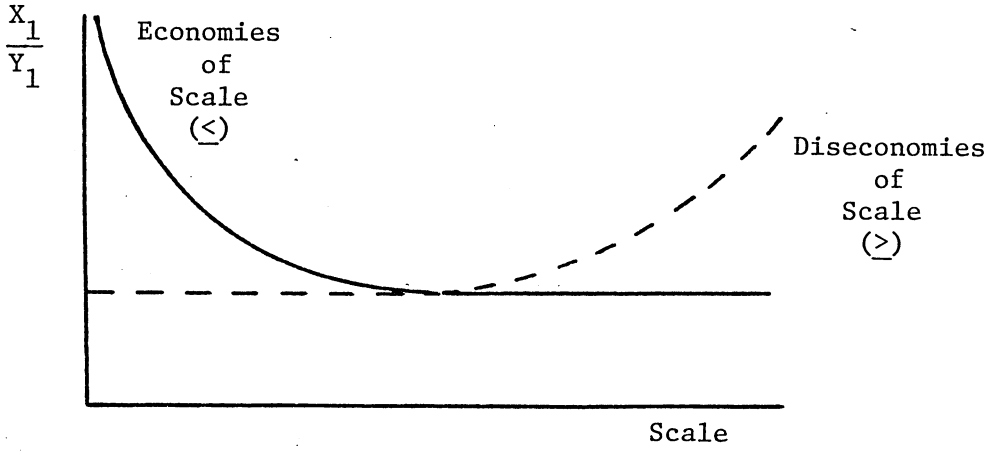
$$z_t \geq 0 \qquad t = 1, \ldots, T.$$

The input constraints and the output constraints are identical except that the inequality sign is reversed. This reversal is consistent with the differences in sign used in the graphic illustration shown in Figure 3. The constraints for the quality variables are considerably different from the input and output variable constraints. The input and output constraints are structured in a form that allows large firms to have nonzero z solution values when a small firm is on the right-hand side of the constraints and vice versa. It is the ratios between all the inputs and all the outputs that are important and not the actual levels of the inputs and outputs. The magnitudes of the z's will adjust for the differences in the input and output levels. The quality and scale constraints, however, must be of a different form to correctly construct the production surface. If a high quality firm is on the right-hand side of the constraints, only high quality firms (on the average) should have nonzero z's in the solution. Since the magnitude of the z's will depend on the input and output levels of the firms, the weighted average form of the quality and scale constraints is necessary. For inputs and outputs, the ratios of the variables are important in the determination of the production surface and not their levels. For the quality and scale variables, it is the levels and not the ratios with the inputs and outputs that are important.

The quality dimensions are defined in such a way that they are like outputs in the sense that they use resources. That is, a firm producing a higher quality output will need to use more (or at least as much) of the inputs. Therefore, the inequality signs for the quality constraints are of the same direction as the output constraints.

Because of the existence of diseconomies as well as economies of scale, it is possible to specify the scale constraint with the inequality in either direction. The economies of scale portion of the relationship shown in Figure 6 as an illustration can be determined with the scale constraint specified with a $\leq$ inequality. The diseconomies of scale portion can be determined with the $\geq$ scale constraint. It is also possible to use a strict equality constraint to force the scale to be exactly the same (on the average) as the scale of the firm on the right-hand side. If the convexity assumption mentioned earlier is not restrictive, it is better to use the inequality constraints for determining the production surface. With the inequality constraints the surface will obviously be convex and therefore the relationships, such as shown in Figure 6, will be smooth curves ranging from a straight line (no economies or diseconomies of scale) to a "U" shaped curve (both economies and diseconomies of scale).

Figure 6: Illustration of the Scale Constraint



The linear programming model specified above yields a direct measure of technical efficiency for multiple input-output production processes. The measure of short-run technical efficiency for firm n is given by $1.0/\sum_{t=1}^{T} z_t$ when the LP model is solved with firm n on the right-hand side of the constraints and the scale constraint is included. If $\sum_{t=1}^{T} z_t = 1.0$, firm n is technically efficient. If firm n is inefficient, then $\sum_{t=1}^{T} z_t$ will be greater than 1.0. The measure of long-run technical efficiency for firm n is exactly the same but the

LP should be solved with the scale constraint omitted. The appropriate measure of economic efficiency will be developed in the following section.

This computational methodology is useful for determining portions of the production surface in addition to calculating efficiency indices for individual firms. A slight change in the formulation of the above LP is useful for easily calculating the production function relationships between alternative input, output, quality, and scale variables. The required modification is to place one of the input or output variables into the objective function. To keep the quality and scale constraints in a proper form it is also necessary to multiply both sides of those constraints by the variable included in the objective function. The input and output constraints remain in the same form as in the previous model. An illustration of the resulting, reformulated model is shown in Figure 7 with one of the output variables in the objective function.

It should be noted that the choice of the variable appearing in the linear programming objective function (Y_{rt} in the example in Figure 7) depends on the information that is desired. The distance being maximized (or minimized in the case of an input variable) is parallel to the axis of the variable in the objective function. It should be stressed that this is <u>not</u> comparable to the choice of the dependent variable in a regression equation, where the results can be drastically different depending on the variable selected. With the linear programming approach, the results are always consistent regardless of the direction towards the frontier surface that the results are generated.

In addition to using observed firms on the right-hand side of the constraints, hypothetical firms can be constructed and used in the LP model as well. This procedure makes it possible to more systematically analyze the frontier relationships between different variables. If an output variable is in the objective function and the right-hand side value of an input constraint is varied, a frontier productivity curve is traced out. A frontier transformation curve results if an output constraint is varied with an output variable in the objective function. With an input variable in the objective function and varying an input constraint, a frontier isoquant is computed. If an input variable is used in the objective function, the problem is then one of minimization rather than maximization as shown in Figure 7.

Figure 7: The LP Computational Model

$$\begin{array}{lll}
\text{Maximize} & \sum_{t=1}^{T} z_t Y_{rt} & \\
\text{Subject to:} & \sum_{t=1}^{T} z_t X_{it} \le X_{is} & i = 1, \ldots, I \\
 & \sum_{t=1}^{T} z_t Y_{jt} \ge Y_{js} & j = 1, \ldots, J \\
 & \sum_{t=1}^{T} z_t Y_{rt} \, (Q_{kt} - Q_{ks}) \ge 0 & k = 1, \ldots, K \\
 & \sum_{t=1}^{T} z_t Y_{rt} \, (S_t - S_s) \left[\begin{matrix} \le \\ = \\ \ge \end{matrix}\right] 0 & \\
 & z_t \ge 0 & t = 1, \ldots, T.
\end{array}$$

Least-Cost Modification

The basic computational algorithm as described above can be modified to find the least-cost method of producing given levels of outputs with specified firm characteristics from the observed data. Letting C_i = the unit price of input i, the least-cost algorithm is:

$$\text{Minimize} \quad \sum_{t=1}^{T} z_t \sum_{i=1}^{I} C_i X_{it}$$

$$\text{Subject to:} \quad \sum_{t=1}^{T} z_t Y_{jt} \geq Y_{js} \qquad j = 1, \ldots, J$$

$$\sum_{t=1}^{T} z_t \sum_{i=1}^{I} C_i X_{it} \; (Q_{kt} - Q_{ks}) \geq 0 \qquad k = 1, \ldots, K$$

$$\sum_{t=1}^{T} z_t \sum_{i=1}^{I} C_i X_{it} \; (S_t - S_s) \begin{bmatrix} \leq \\ = \\ \geq \end{bmatrix} 0$$

$$z_t \geq 0 \qquad t = 1, \ldots, T.$$

Verbally, the problem is to minimize the total cost of production subject to the constraints that the constructed firm has at least as much of each specified output and equals or exceeds the various firm quality constraints.

From the solution values of the z_t's, the cost-minimizing level of each input is given by $X_i^* = \sum_{t=1}^{T} z_t X_{it}$. If certain inputs are considered fixed, they can be included as constraints in the LP model, either as equalities or inequalities if idle capacity is allowed, and enter the objective function only as fixed constants.

This procedure allows the computation of least-cost methods of producing various output combinations with specified firm quality factors, given input prices, and the production relationships observed from the cross-section of firms. Instead of minimizing with respect to one input (or maximizing with respect to one output) as done in the basic computational approach, all the inputs are weighted by their unit prices, and their weighted sum is minimized. A measure of total economic efficiency results from the solution of the model in this formulation. For firm n the measure of economic efficiency would be:

$$EE_n = \frac{\sum_{t=1}^{T} z_t \sum_{i=1}^{N} C_i X_{it}}{\sum_{i=1}^{N} C_i X_{in}} \; .$$

Alternatively, the quantity $\sum_{i=1}^{T} C_i X_{it}$ in the above formulation can be replaced by the actual total expenditures of the t^{th} firm. This approach also yields information about the cost-minimizing behavior observed for the sample of firms. These procedures make the appropriate link between the production relationships and the cost relationships for this type of frontier analysis. Revenue maximization problems with given levels of inputs can also be formulated and solved by this algorithm. The same procedure as outlined above could be used with unit prices of the outputs used instead of input prices.

The Linear-Programming Computer Program:

As shown in Figure 7, the computational procedure for calculating efficiency measures and frontier multi-product, multi-factor production and cost relationships results in a very simple linear programming problem. Since most empirical applications of the frontier technique involve solving a large number of linear programs, it is desirable to have a general computer program that reads the production and cost data for each firm, reads several control parameters describing a specific application, processes all of this input information, sets up the linear programming problems, and uses an LP algorithm as a subroutine to generate the desired solutions.

The remaining sections of this paper describe and illustrate the use of the computer program developed for calculating efficiency measures and frontier production and cost relationships. The FØRTRAN computer program is listed in the final section of this paper. The alternative modes of operating the program are described, the control parameters are defined, the input deck structure is laid out, and several illustrative runs of the program are presented in the following sections.

Modes of Operation:

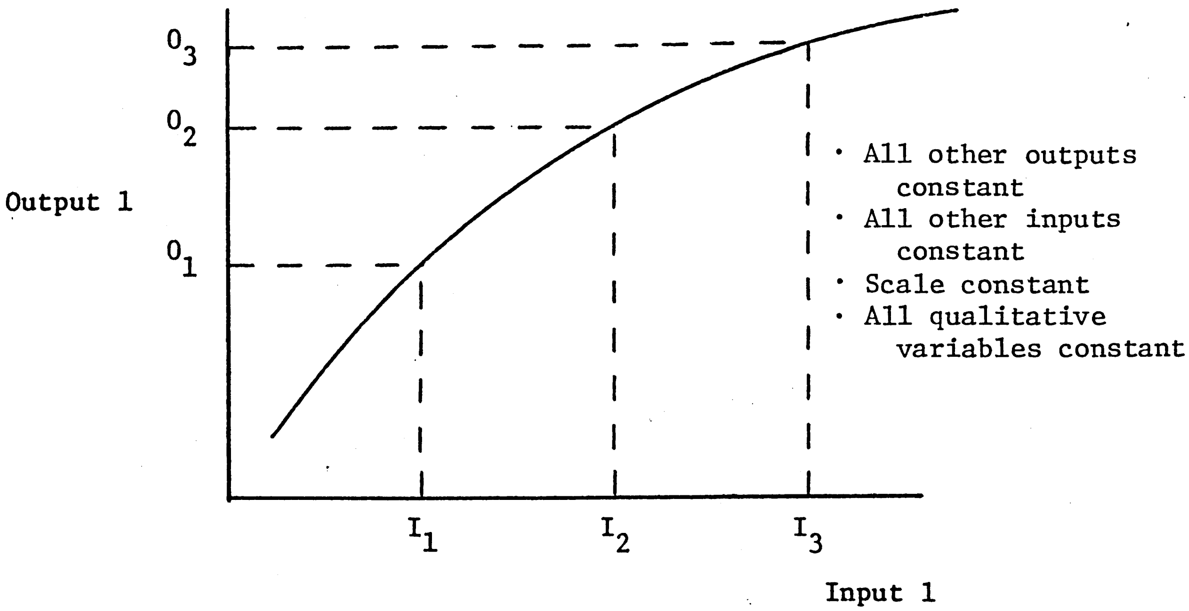
Given the large number of alternative measures of efficiency that have been defined and the alternative portions of the production surface that can be calculated, there are several different ways in which the computer program can be run. Each of the available options are described below.

General efficiency option: This option calculates the general multi-product, multi-factor technical efficiency index for each firm in the sample. The LP solution value yields the information required to directly calculate the efficiency index. The LP solution value will equal -1.0 for firms on the production frontier surface and will be less than -1.0 for firms not on the surface. Since the LP activities are firms in the sample, the primal solution also indicates which frontier firms, when added together with the optimal coefficients of the primal variables as weights, dominate each of the nonfrontier firms.

Specific efficiency option: This option calculates the technical efficiency index relative to a specified variable for each firm in the sample. The LP solution value yields the optimal value of the specified variable (minimum value for inputs, maximum value for outputs) with all of the other variables constrained for the particular firm. For input variables, the technical efficiency index is calculated by dividing the firm's actual value for the specified input by the LP solution value. For input variables, the technical efficiency index is calculated by dividing the LP solution value by the firm's actual value for the specified output.

Production surface option: This option calculates points on a portion of the production frontier surface. For example, if the frontier relationship between one input and one output is desired the procedure would be to specify the output variable in the objective function and to solve the LP for alternative values on the RHS of the input constraint. As illustrated in Figure 8, the program computes the maximum amount of output 1 for each of the levels of input 1 (I_1, I_2, and I_3) with all the other outputs, inputs, and qualitative

Figure 8: Calculation of the Production Surface


0_3
0_2
Output 1
0_1
• All other outputs constant
• All other inputs constant
• Scale constant
• All qualitative variables constant
I_1
I_2
I_3
Input 1

factors held constant. The three points on the frontier production surface (0_1, 0_2, and 0_3) are calculated in one replication of the program although the replication requires three linear programming problems to be solved. If an input is specified in the objective function and the RHS of an input constraint is varied, an isoquant will be traced out. If an output is specified in the objective function and the RHS of an output constraint is varied, a product transformation curve will be traced out. If an output (input) is specified in the objective function and the RHS of an input (output) constraint is varied, a marginal productivity curve is traced out. Similarly, outputs or inputs can be specified in the objective function and the RHS of a quality factor or a scale constraint varied to yield frontier relationships for the quality and scale factors.

Variable mean option: This option is designed to enable the user to specify the point at which the production surface is to be calculated. If this option is not used, the RHS of all of the constraints are set equal to the mean value for each of the respective variables. The production surface option enables the user to then vary one of the constraint RHS's. With this variable mean option, it is possible to change all or several of the constraint variables in order to trace out different portions of the multi-dimensional production surface.

Least-cost option: This option is designed to trace out a portion of the frontier cost surface using the per unit cost approach rather than total expenditures directly. Per unit costs must be inputted for each input variable that is to be included in total cost for the objective function. Except for the additional data requirement, this option has the same procedures and capabilities as the production surface option. It is also possible to use this option along with the specific efficiency option to calculate indices of total economic efficiency for each firm in the sample.

Control Parameters:

For any run of the computer program, the user must specify several parameters that control the way the LP problem is to be set up. These parameters allow the user considerable flexibility in using the program under the alternative options described above. The control parameters are listed and defined below:

NVAR = Number of constraint variables in the problem.

ISO = 1 if a production surface run; = 0 if an efficiency run.

IEF = 1 if a general efficiency run; = 0 if a specific efficiency run.

IVM = 1 if different variable means are to be read in for each replication; = 0 otherwise.

ICST = 1 if a "least-cost" run is desired; = 0 otherwise.

ICN = Number of total variables for "least-cost" run; = 0 if ICST = 0.

NREP = Number of replications; = 1 if ISO = 0.

NEQ = Number of equality constraints. The equality constraint variables must be specified first in INDEX (•).

INDEX (•) = Index numbers for variables; objective function variable listed last.

COST (•) = Per unit costs of inputs for "least-cost" option; omit if ICST = 0.

VMST (•) = Variable means; listed in same order as INDEX; omit if IVM = 0.

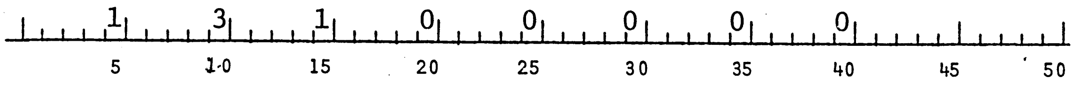
IFRST = Index number of "isoquant" variable; omit if ISO = 0.

VONE (•) = Three alternative values of "isoquant" variables; omit if ISO = 0.

Input Deck Structure:

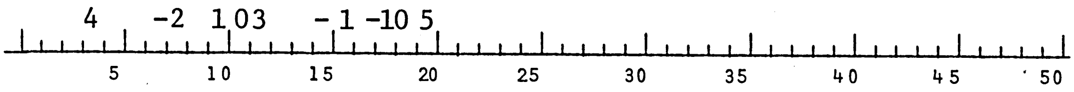
The sequence of control parameter cards and data cards are described below:

CARD 1: NREP, NVAR, ISO, IEF, NEQ, ICST, IVM, ICN (Mandatory)



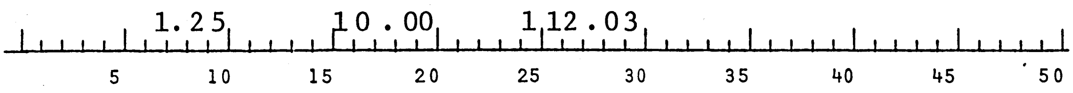
The format for Card 1 is 8I5.

CARD 2: INDEX (•) (Mandatory)



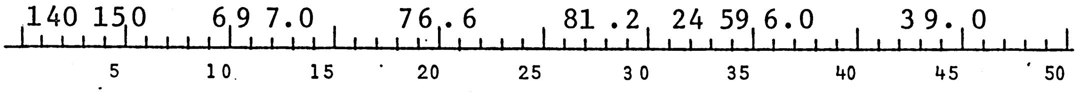
The index indicates the position of the desired variable in VST, the input data matrix (see below). A minus sign indicates that the variable is used in a $\geq$ constraint and a positive sign indicates that the variable is used in a $\leq$ constraint. Qualitative variables are specified by adding 100 to the basic index and therefore 103 represents a quality variable that appears in the 3rd position of VST. The format for Card 2 is 20I4.

CARD 3: COST (•) (Optional)



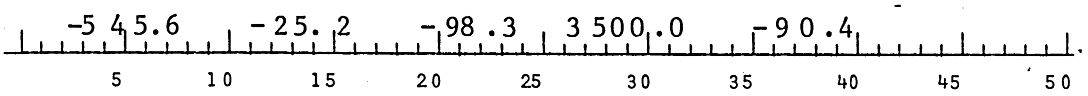
This card(s) should be included only if ICST = 1. The format is 8F10.2 and if there are more than 8 inputs, use more than one card.

CARD 4: IDENT (•), (VST(I,•), I = 1, 20). (Mandatory)



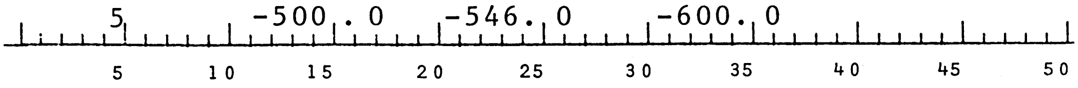
These cards should be set up in a format most convenient for reading in the data to be used in the analysis and the READ and FORMAT statement in the program should be appropriately modified. IDENT (•) should contain an identification number (integer) for each firm included in the analysis and VST (I,•) should contain a value (real number) for each of the I variables and for each firm. The program is currently set up so that an IDENT code of '999999' indicates that all of the data has been read in. The current format is I6, 9F10.0.

CARD 5: VMST (•) (Optional)



This card(s) should be included only if IVM = 1. The format is 10F8.0 and if NVAR > 10 more than one card must be used. Note that the sign of the mean values must correspond to the sign of the appropriate variable in INDEX (•).

CARD 6: IFRST, (VONE(I), I = 1, 3) (Optional)



This card(s) should be included only if ISO = 1. The format is (I5, 5X, 3F10.0) and there should be NREP of these cards if ISO = 1. The first integer on the card refers to the position in INDEX of the variable to be varied over three values and the following three real numbers are the values.

Illustrative Uses of the Program:

To better illustrate the alternative uses of this computer program, several runs are described, the input cards laid out, and the output presented in this section. For these examples, variables 1 and 2 are outputs, variables 3 and 4 are inputs, variable 5 is a scale measure, and variable 6 is a qualitative factor. There are 50 firms in the sample.

Example 1: A general efficiency run:

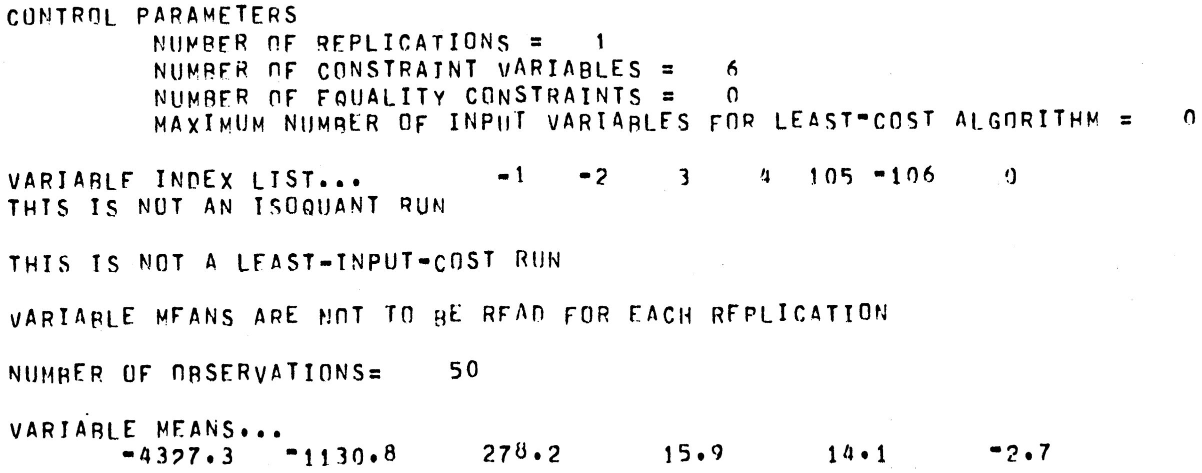
CARD 1: 1, 6, 0, 1, 0, 0, 0, 0

CARD 2: -1, -2, 3, 4, 105, -106

CARD(s) 4: DATA

The output of this run is shown on the following pages for the first 3 firms in the sample. The general technical efficiency indices calculated from the results are:

Firm	Solution	Index (-1.0/Solution)
1	-1.00	1.000
2	-4.78	0.209
3	-1.97	0.507



```
CONTROL PARAMETERS
        NUMBER OF REPLICATIONS =    1
        NUMBER OF CONSTRAINT VARIABLES =    6
        NUMBER OF FQUALITY CONSTRAINTS =    0
        MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM =    0

VARIABLF INDEX LIST...        -1    -2     3     4   105  -106     0
THIS IS NOT AN ISOQUANT RUN

THIS IS NOT A LFAST-INPUT-COST RUN

VARIABLE MFANS ARE NOT TO BE RFAD FOR EACH RFPLICATION

NUMBER OF OBSERVATIONS=     50

VARIABLE MEANS...
     -4327.3   -1130.8      278.2       15.9       14.1       -2.7
```

LISTING OF DATA MATRIX

```
  1   1002    -140.0   185.8    0.0    7.5   -4.0   -2236.0
  2   1052   -2490.0   560.0   74.0   86.6   -3.0   -3743.0
  3   1057   -1068.0   232.2    0.0   27.8   -3.0   -4009.0
  4   1089    -113.0   111.5    0.0    4.9   -2.0   -2518.0
  5   1090    -937.0   281.5   30.4   12.5   -3.0   -5202.0
  6   1101   -1699.0   172.0    0.0    5.2   -3.0   -3039.0
  7   1107    -166.0   113.3    0.0    4.6   -2.0   -1716.0
  8   1345    -273.0   122.1    5.3    6.4   -2.0   -2063.0
  9   1353    -139.0   110.0    0.0    5.9   -2.0   -2262.0
 10   1360   -3832.0   258.2    0.0   11.3   -4.0   -4920.0
 11   1365    -970.0   288.3    0.0   11.4   -3.0   -4207.0
 12   1378   -2447.0   469.0    0.0   16.2   -2.0   -7324.0
 13   1380    -598.0   183.8    0.0    5.8   -2.0   -2676.0
 14   1480    -317.0   256.0    0.0   18.0   -4.0   -4024.0
 15   1481    -883.0   248.6   14.0   18.0   -4.0   -3637.0
 16   1546    -536.0   120.0    0.0    3.3   -2.0   -2176.0
 17   1552   -1456.0   113.0    0.0    4.1   -2.0   -1530.0
 18   1561   -1521.0   116.3    0.0    4.8   -2.0   -1985.0
 19   1572    -314.0   302.0   17.0   11.5   -4.0   -4895.0
 20   1573    -290.0   100.5    0.0    4.6   -2.0   -1831.0
 21   1574   -5466.0   700.7   40.2   31.3   -2.0   -5382.0
 22   1590    -256.0   109.6    0.0    5.6   -3.0   -2278.0
 23   1599    -470.0   198.2    3.0    7.1   -2.0   -2963.0
 24   1601    -589.0   282.0    0.0   10.7   -2.0   -4192.0
 25   1616   -2285.0   320.0    0.0   13.0   -2.0   -4598.0
 26   1620    -730.0   275.0   39.0   17.8   -3.0   -4049.0
 27   1674    -134.0   507.0   25.1   25.8   -3.0   -7692.0
 28   1759   -2378.0   553.4   97.0   37.1   -3.0   -7274.0
 29   1808    -769.0    78.4    0.0    3.5   -2.0   -1855.0
 30   1812   -1722.0   114.0    0.0    4.6   -1.0   -1998.0
 31   1815   -1903.0   115.0    0.0   10.8   -2.0   -2080.0
 32   1816   -1952.0   121.0    0.0    5.6   -2.0   -2115.0
 33   1890    -626.0   499.9   12.9   22.7   -2.0   -7443.0
 34   1915    -380.0   222.7   10.1    9.1   -2.0   -3903.0
 35   1926    -317.0   221.5    7.6   10.5   -4.0   -3878.0
 36   1927    -304.0   311.5    9.0   12.9   -3.0   -4901.0
 37   1949   -1159.0   188.0    2.0    6.8   -2.0   -3444.0
 38   1950   -4410.0   537.9    0.0   22.4   -3.0   -6946.0
 39   1963    -717.0   474.0   55.0   19.2   -4.0   -8267.0
 40   1976    -486.0   268.0   30.0   17.4   -4.0   -5056.0
 41   1977    -768.0   363.6   28.3   15.5   -3.0   -5258.0
 42   2002   -1067.0   470.0   42.0   21.7   -3.0   -8737.0
 43   2006    -239.0   258.0    0.0    8.3   -2.0   -3572.0
 44   2008    -624.0   370.0   69.0   18.0   -4.0   -6380.0
 45   2015   -2949.0   449.9   73.1   17.3   -3.0   -8479.0
 46   2017    -645.0   285.5    0.0    9.5   -3.0   -4453.0
 47   2020   -1067.0   350.7   67.2   15.0   -4.0   -6669.0
 48   2024    -463.0   252.6   11.1    9.9   -3.0   -4641.0
 49   2031   -1474.0   470.0   30.0   18.6   -4.0   -8824.0
 50   2184       0.0   200.0    0.5    6.9   -2.0   -3043.0
```

```
ITERATIONS=  12

SOLUTION VALUE=    -.100000E+01

     PRIMAL VARIABLES          DUAL SLACK                  COST SENSITIVITY
  1        .100000E+01         0.                      -.318872E+00          .161400E+00

     DUAL VARIABLES          PRIMAL SLACK               RESOURCE SENSITIVITY
  1       0.                   0.                      0.                    .100000E+21
  2       0.                   0.                      0.                    .100000E+21
  3       -.538358E-02         0.                      -.185750E+03        0.
  4       0.                   0.                      0.                    .100000E+21
  5       0.                   0.                      0.                    .100000E+21
  6       -.409960E+00         0.                      -.100000E+21        0.

              1        1002             1.000

ITERATIONS=  12

SOLUTION VALUE=    -.477778E+01

     PRIMAL VARIABLES          DUAL SLACK                  COST SENSITIVITY
 10        .309828E+00         0.                      -.107062E+01          .145835E+00
 22        .415812E+01         0.                      -.786527E-01          .150731E+00
 29        .309828E+00         0.                      -.354967E+00          .314385E-01

     DUAL VARIABLES          PRIMAL SLACK               RESOURCE SENSITIVITY
  1       0.                    .782829E+04            -.782829E+04          .100000E+21
  2       -.280640E-03         0.                      -.516603E+04          .118197E+04
  3       -.977960E-02         0.                      -.374214E+03          .506031E+03
  4       0.                    .740000E+02            -.740000E+02          .100000E+21
  5       0.                    .385906E+03            -.385906E+03          .100000E+21
  6       -.449483E+00         0.                      -.366054E+00          .201712E+01

              2        1052             0.209

ITERATIONS=  21

SOLUTION VALUE=    -.197110E+01

     PRIMAL VARIABLES          DUAL SLACK                  COST SENSITIVITY
 10        .137784E+00         0.                      -.107062E+01          .145835E+00
 22        .169553E+01         0.                      -.786527E-01          .150731E+00
 29        .137784E+00         0.                      -.354967E+00          .314385E-01

     DUAL VARIABLES          PRIMAL SLACK               RESOURCE SENSITIVITY
  1       0.                    .786906E+03            -.786906E+03          .100000E+21
  2       -.280640E-03         0.                      -.210652E+04          .525635E+03
  3       -.977960E-02         0.                      -.376163E+02          .225038E+03
  4       0.                   0.                      0.                    .100000E+21
  5       0.                    .432517E+02            -.432517E+02          .100000E+21
  6       -.449483E+00         0.                      -.162788E+00          .897035E+00

              3        1057             0.507
```

Example 2: A specific efficiency run:

CARD 1: 1, 5, 0, 0, 0, 0, 0, 0

CARD 2: -2, 3, 4, 105, -106, -1

CARD(s) 4: DATA

The output of this run is shown on the following pages for the first 3 firms in the sample. The calculated technical efficiency indices relative to variable 1 (an output) are:

Firm	Actual Value	Frontier Value	Index (-Actual/Frontier)
1	2,236	- 2,236	1.000
2	3,743	-11,824	0.317
3	4,009	- 4,903	0.818

```
CONTROL PARAMETERS
        NUMBER OF REPLICATIONS =    1
        NUMBER OF CONSTRAINT VARIABLES =    5
        NUMBER OF EQUALITY CONSTRAINTS =    0
        MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM = 0

VARIABLE INDEX LIST...          -2     3     4   105 -106    -1
THIS IS NOT AN ISOQUANT RUN

THIS IS NOT A LEAST-INPUT-COST RUN

VARIABLE MEANS ARE NOT TO BE READ FOR EACH REPLICATION

NUMBER OF OBSERVATIONS=     50

VARIABLE MEANS...
       -1130.8      278.2       15.9        14.1        -2.7    -4327.3
```

LISTING OF DATA MATRIX

```
  1    1002    -2236.0    -140.0    185.8     0.0     7.5    -4.0    -1.0
  2    1052    -3743.0   -2490.0    560.0    74.0    86.6    -3.0    -1.0
  3    1057    -4009.0   -1068.0    232.2     0.0    27.8    -3.0    -1.0
  4    1089    -2518.0    -113.0    111.5     0.0     4.9    -2.0    -1.0
  5    1090    -5202.0    -937.0    281.5    30.4    12.5    -3.0    -1.0
  6    1101    -3039.0   -1699.0    172.0     0.0     5.2    -3.0    -1.0
  7    1107    -1716.0    -166.0    113.3     0.0     4.6    -2.0    -1.0
  8    1345    -2063.0    -273.0    122.1     5.3     6.4    -2.0    -1.0
  9    1353    -2262.0    -139.0    110.0     0.0     5.9    -2.0    -1.0
 10    1360    -4920.0   -3832.0    258.2     0.0    11.3    -4.0    -1.0
 11    1365    -4207.0    -970.0    288.3     0.0    11.4    -3.0    -1.0
 12    1378    -7324.0   -2447.0    469.0     0.0    16.2    -2.0    -1.0
 13    1380    -2676.0    -598.0    183.8     0.0     5.8    -2.0    -1.0
 14    1480    -4024.0    -317.0    256.0     0.0    18.0    -4.0    -1.0
 15    1481    -3637.0    -883.0    248.6    14.0    18.0    -4.0    -1.0
 16    1546    -2176.0    -536.0    120.0     0.0     3.3    -2.0    -1.0
 17    1552    -1530.0   -1456.0    113.0     0.0     4.1    -2.0    -1.0
 18    1561    -1985.0   -1521.0    116.3     0.0     4.8    -2.0    -1.0
 19    1572    -4895.0    -314.0    302.0    17.0    11.5    -4.0    -1.0
 20    1573    -1831.0    -290.0    100.5     0.0     4.6    -2.0    -1.0
 21    1574    -5382.0   -5466.0    700.7    40.2    31.3    -2.0    -1.0
 22    1590    -2278.0    -256.0    109.6     0.0     5.6    -3.0    -1.0
 23    1599    -2963.0    -470.0    198.2     3.0     7.1    -2.0    -1.0
 24    1601    -4192.0    -589.0    282.0     0.0    10.7    -2.0    -1.0
 25    1616    -4598.0   -2285.0    320.0     0.0    13.0    -2.0    -1.0
 26    1620    -4049.0    -730.0    275.0    39.0    17.8    -3.0    -1.0
 27    1674    -7692.0    -134.0    507.0    25.1    25.8    -3.0    -1.0
 28    1759    -7274.0   -2378.0    553.4    97.0    37.1    -3.0    -1.0
 29    1808    -1855.0    -769.0     78.4     0.0     3.5    -2.0    -1.0
 30    1812    -1998.0   -1722.0    114.0     0.0     4.6    -1.0    -1.0
 31    1815    -2080.0   -1903.0    115.0     0.0    10.8    -2.0    -1.0
 32    1816    -2115.0   -1952.0    121.0     0.0     5.6    -2.0    -1.0
 33    1890    -7443.0    -626.0    499.9    12.9    22.7    -2.0    -1.0
 34    1915    -3903.0    -380.0    222.7    10.1     9.1    -2.0    -1.0
 35    1926    -3878.0    -317.0    221.5     7.6    10.5    -4.0    -1.0
 36    1927    -4901.0    -304.0    311.5     9.0    12.9    -3.0    -1.0
 37    1949    -3444.0   -1159.0    188.0     2.0     6.8    -2.0    -1.0
 38    1950    -6946.0   -4410.0    537.9     0.0    22.4    -3.0    -1.0
 39    1963    -8267.0    -717.0    474.0    55.0    19.2    -4.0    -1.0
 40    1976    -5056.0    -486.0    268.0    30.0    17.4    -4.0    -1.0
 41    1977    -5258.0    -768.0    363.6    28.3    15.5    -3.0    -1.0
 42    2002    -8737.0   -1067.0    470.0    42.0    21.7    -3.0    -1.0
 43    2006    -3572.0    -239.0    258.0     0.0     8.3    -2.0    -1.0
 44    2008    -6380.0    -624.0    370.0    69.0    18.0    -4.0    -1.0
 45    2015    -8479.0   -2949.0    449.9    73.1    17.3    -3.0    -1.0
 46    2017    -4453.0    -645.0    285.5     0.0     9.5    -3.0    -1.0
 47    2020    -6669.0   -1067.0    350.7    67.2    15.0    -4.0    -1.0
 48    2024    -4641.0    -463.0    252.6    11.1     9.9    -3.0    -1.0
 49    2031    -8824.0   -1474.0    470.0    30.0    18.6    -4.0    -1.0
 50    2184    -3043.0       0.0    200.0     0.5     6.9    -2.0    -1.0
```

```
ITERATIONS=  19

SOLUTION VALUE=    -.223600E+04

      PRIMAL VARIABLES          DUAL SLACK               COST SENSITIVITY
  1        .100000E+01          0.                   -.942017E+03          .615227E+03

      DUAL VARIABLES          PRIMAL SLACK             RESOURCE SENSITIVITY

  1       0.                    0.                   0.                    .100000E+21
  2       -.120377E+02          0.                   -.185750E+03        0.
  3       0.                    0.                   0.                    .100000E+21
  4       -.230797E+00          0.                   -.968817E+04        0.
  5       -.853991E+00          0.                   -.100000E+21        0.

            1       1002             1.000

ITERATIONS=  12

SOLUTION VALUE=    -.118245E+05

      PRIMAL VARIABLES          DUAL SLACK               COST SENSITIVITY
 10        .120168E+01          0.                   -.119184E+04          .116718E+02
 29        .318721E+01          0.                   -.100000E+21          .580863E+02

      DUAL VARIABLES          PRIMAL SLACK             RESOURCE SENSITIVITY
  1       0.                       .456581E+04       -.456581E+04          .100000E+21
  2       -.211153E+02          0.                   -.362376E+03          .100000E+21
  3       0.                       .740000E+02       -.740000E+02          .100000E+21
  4       0.                       .936607E+06       -.936607E+06          .100000E+21
  5       -.108036E+00          0.                   -.106716E+05          .132568E+05

            2       1052             0.317

ITERATIONS=  19

SOLUTION VALUE=    -.490296E+04

      PRIMAL VARIABLES          DUAL SLACK               COST SENSITIVITY
 10        .498269E+00          0.                   -.119184E+04          .116718E+02
 29        .132155E+01          0.                   -.100000E+21          .580863E+02

      DUAL VARIABLES          PRIMAL SLACK             RESOURCE SENSITIVITY
  1       0.                       .185764E+04       -.185764E+04          .100000E+21
  2       -.211153E+02          0.                   -.147436E+03          .100000E+21
  3       0.                    0.                   0.                    .100000E+21
  4       0.                       .100014E+06       -.100014E+06          .100000E+21
  5       -.108036E+00          0.                   -.442491E+04          .549682E+04

            3       1057             0.818
```

Example 3: A production surface run:

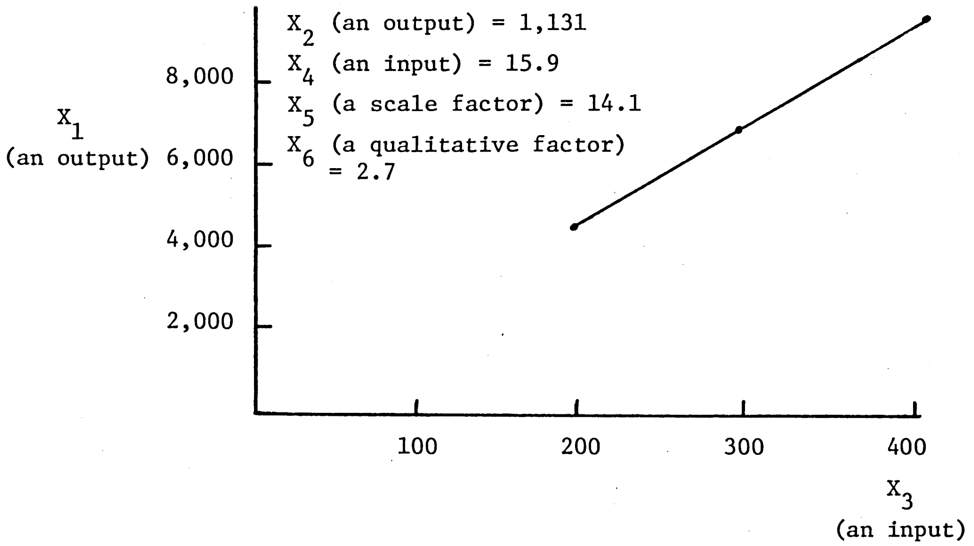
CARD 1: 1, 5, 1, 0, 0, 0, 0, 0

CARD 2: -2, 3, 4, 105, -106, -1

CARD(s) 4: DATA

CARD 6: 2, 200.0, 300.0, 400.0

The output of this run is shown on the following pages. The resulting portion of the frontier production surface is illustrated below.



```
CONTROL PARAMETERS
        NUMBER OF REPLICATIONS =    1
        NUMBER OF CONSTRAINT VARIABLES =    5
        NUMBER OF EQUALITY CONSTRAINTS =    0
        MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM = 0

VARIABLE INDEX LIST...          -2     3     4   105  -106    -1
THIS IS AN ISOQUANT RUN

THIS IS NOT A LEAST-INPUT-COST RUN

VARIABLE MEANS ARE NOT TO BE READ FOR EACH REPLICATION

NUMBER OF OBSERVATIONS=     50

VARIABLE MEANS...
      -1130.8      278.2       15.9        14.1        -2.7   -4327.3
```

The listing of the data matrix for this example is exactly the same as the listing for the previous run.

```
ISO-RUN VARIABLE INDICATOR AND VALUES
          2            200.        300.        400.
ITERATIONS=  18

SOLUTION VALUE=    -.434511E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .326766E+00         0.                  -.119184E+04        .120126E+02
 29       .147570E+01         0.                  -.100000E+21        .594890E+02

     DUAL VARIABLES         PRIMAL SLACK           RESOURCE SENSITIVITY
  1      0.                    .125622E+04        -.125622E+04        .100000E+21
  2      -.217255E+02         0.                  -.105256E+03        .100000E+21
  3      0.                    .158538E+02        -.158538E+02        .100000E+21
  4      0.                    .334242E+05        -.334242E+05        .100000E+21
  5      -.111159E+00         0.                  -.480223E+04        .350357E+04
ITERATIONS=  17

SOLUTION VALUE=    -.651766E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .490149E+00         0.                  -.119184E+04        .120126E+02
 29       .221354E+01         0.                  -.100000E+21        .594890E+02

     DUAL VARIABLES         PRIMAL SLACK           RESOURCE SENSITIVITY
  1      0.                    .244971E+04        -.244971E+04        .100000E+21
  2      -.217255E+02         0.                  -.205256E+03        .100000E+21
  3      0.                    .158538E+02        -.158538E+02        .100000E+21
  4      0.                    .501363E+05        -.501363E+05        .100000E+21
  5      -.111159E+00         0.                  -.720335E+04        .525536E+04
ITERATIONS=  18

SOLUTION VALUE=    -.869021E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .653532E+00         0.                  -.119184E+04        .120126E+02
 29       .295139E+01         0.                  -.100000E+21        .594890E+02

     DUAL VARIABLES         PRIMAL SLACK           RESOURCE SENSITIVITY
  1      0.                    .364320E+04        -.364320E+04        .100000E+21
  2      -.217255E+02         0.                  -.305256E+03        .100000E+21
  3      0.                    .158538E+02        -.158538E+02        .100000E+21
  4      0.                    .668484E+05        -.668484E+05        .100000E+21
  5      -.111159E+00         0.                  -.960446E+04        .700715E+04
```

Example 4: A production surface run with variable means:

CARD 1: 2, 5, 1, 0, 0, 0, 1, 0

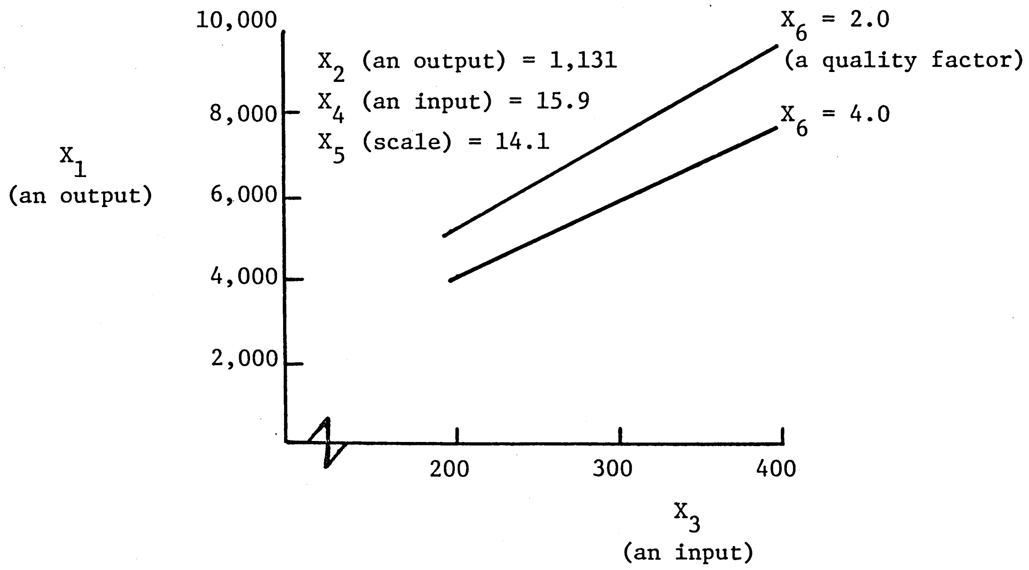
CARD 2: -2, 3, 4, 105, -106, -1

CARD(s) 4: DATA

first replication
CARD 5: -1130.8, 278.2, 15.9, 14.1, -2.0
CARD 6: 2, 200.0, 300.0, 400.0

second replication
CARD 5: -1130.8, 278.2, 15.9, 14.1, -4.0
CARD 6: 2, 200.0, 300.0, 400.0

The data listing for this run is exactly the same as the listing for example 2. The output of this run is shown on the following pages. The resulting portions of the frontier production surface are illustrated below.



```
CONTROL PARAMETERS
         NUMBER OF REPLICATIONS =    2
         NUMBER OF CONSTRAINT VARIABLES =    5
         NUMBER OF EQUALITY CONSTRAINTS =    0
         MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM =0

VARIABLE INDEX LIST...          -2     3     4   105  -106    -1
THIS IS AN ISOQUANT RUN

THIS IS NOT A LEAST-INPUT-COST RUN

VARIABLE MEANS ARE TO BE READ FOR EACH REPLICATION

NUMBER OF OBSERVATIONS=     50

VARIABLE MEANS...
       -1130.8      278.2       15.9       14.1       -2.7   -4327.3
```

```
VARIABLE MEANS FOR THIS REPLICATION
      -1130.8      278.2       15.9        14.1       -2.0        0.0

ISO-RUN VARIABLE INDICATOR AND VALUES
              2             200.        300.       400.
ITERATIONS=  16

SOLUTION VALUE=    -.473456E+04

      PRIMAL VARIABLES         DUAL SLACK               COST SENSITIVITY
 29        .255232E+01         0.                   -.100000E+21          .638794E+02

      DUAL VARIABLES         PRIMAL SLACK             RESOURCE SENSITIVITY
  1       0.                     .831936E+03        -.831936E+03          .100000E+21
  2       -.236728E+02       0.                     -.847731E+02          .100000E+21
  3       0.                     .159000E+02        -.159000E+02          .100000E+21
  4       0.                     .500588E+05        -.500588E+05          .100000E+21
  5       -.121122E+00       0.                     -.762259E+04         0.
ITERATIONS=  14

SOLUTION VALUE=    -.710184E+04

      PRIMAL VARIABLES         DUAL SLACK               COST SENSITIVITY
 29        .382848E+01         0.                   -.100000E+21          .853993E+02

      DUAL VARIABLES         PRIMAL SLACK             RESOURCE SENSITIVITY
  1       0.                     .181330E+04        -.181330E+04          .100000E+21
  2       -.236728E+02       0.                     -.160000E+02          .100000E+21
  3       0.                     .159000E+02        -.159000E+02          .100000E+21
  4       0.                     .750882E+05        -.750882E+05          .100000E+21
  5       0.                    -.745058E-08         .745058E-08          .100000E+21
ITERATIONS=  15

SOLUTION VALUE=    -.946912E+04

      PRIMAL VARIABLES         DUAL SLACK               COST SENSITIVITY
 29        .510465E+01         0.                   -.100000E+21          .853993E+02

      DUAL VARIABLES         PRIMAL SLACK             RESOURCE SENSITIVITY
  1       0.                     .279467E+04        -.279467E+04          .100000E+21
  2       -.236728E+02       0.                     -.228571E+03          .100000E+21
  3       0.                     .159000E+02        -.159000E+02          .100000E+21
  4       0.                     .100118E+06        -.100118E+06          .100000E+21
  5       0.                     .186265E-06        -.186265E-06          .100000E+21
```

```
VARIABLE MEANS FOR THIS REPLICATION
       -1130.8     278.2      15.9      14.1      -4.0       0.0

ISO-RUN VARIABLE INDICATOR AND VALUES
             2           200.      300.      400.
ITERATIONS=  19

SOLUTION VALUE=   -.381129E+04

     PRIMAL VARIABLES       DUAL SLACK            COST SENSITIVITY
 10       .774653E+00       0.                -.119184E+04       .105237E+02

     DUAL VARIABLES       PRIMAL SLACK          RESOURCE SENSITIVITY
  1       0.                .183767E+04       -.183767E+04       .100000E+21
  2      -.190565E+02     0.                  -.123813E+03       .100000E+21
  3       0.                .159000E+02       -.159000E+02       .100000E+21
  4       0.                .105314E+05       -.105314E+05       .100000E+21
  5      -.975027E-01     0.                  -.252288E+04      0.
ITERATIONS=  26

SOLUTION VALUE=   -.571694E+04

     PRIMAL VARIABLES       DUAL SLACK            COST SENSITIVITY

 10       .116198E+01       0.                -.119184E+04       .105237E+02

     DUAL VARIABLES       PRIMAL SLACK          RESOURCE SENSITIVITY
  1       0.                .332191E+04       -.332191E+04       .100000E+21
  2      -.190565E+02     0.                  -.223813E+03       .100000E+21
  3       0.                .159000E+02       -.159000E+02       .100000E+21
  4       0.                .157971E+05       -.157971E+05       .100000E+21
  5      -.975027E-01     0.                  -.378432E+04      0.
ITERATIONS=  26

SOLUTION VALUE=   -.762259E+04

     PRIMAL VARIABLES       DUAL SLACK            COST SENSITIVITY
 10       .154931E+01       0.                -.119184E+04       .105237E+02

     DUAL VARIABLES       PRIMAL SLACK          RESOURCE SENSITIVITY
  1       0.                .480614E+04       -.480614E+04       .100000E+21
  2      -.190565E+02     0.                  -.323813E+03       .100000E+21
  3       0.                .159000E+02       -.159000E+02       .100000E+21
  4       0.                .210628E+05       -.210628E+05       .100000E+21
  5      -.975027E-01     0.                  -.504577E+04      0.
```

Example 5: A total economic efficiency run:

CARD 1: 1, 4, 0, 0, 0, 1, 0, 6

CARD 2: -1, -2, 105, -106, 3, 4

CARD 3: 20.0, 10.0

CARD(s) 4: DATA

The output of this run is shown on the following pages for the first 3 firms in the sample. The total economic efficiency indices are:

Firm	Actual Cost	Frontier Cost	Index (Frontier/Actual)
1	3,715	3,715	1.000
2	11,940	3,677	0.308
3	4,644	3,753	0.808

```
CONTROL PARAMETERS
          NUMBER OF REPLICATIONS =    1
          NUMBER OF CONSTRAINT VARIABLES =    4
          NUMBER OF EQUALITY CONSTRAINTS =    0
          MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM = 6

VARIABLE INDEX LIST...          -1    -2   105  -106     3
THIS IS NOT AN ISOQUANT RUN

THIS IS A LEAST-INPUT-COST RUN

VARIABLE MEANS ARE NOT TO BE READ FOR EACH REPLICATION

INPUT PRICES FOR COST ANALYSIS
          20.00       10.00
NUMBER OF OBSERVATIONS=     50

VARIABLE MEANS...
        -4327.3    -1130.8        14.1        -2.7       278.2        15.9
```

```
LISTING OF DATA MATRIX

  1    1002    -2236.0    -140.0     7.5    -4.0    3715.0
  2    1052    -3743.0   -2490.0    86.6    -3.0   11940.0
  3    1057    -4009.0   -1068.0    27.8    -3.0    4644.0
  4    1089    -2518.0    -113.0     4.9    -2.0    2230.0
  5    1090    -5202.0    -937.0    12.5    -3.0    5934.0
  6    1101    -3039.0   -1699.0     5.2    -3.0    3440.0
  7    1107    -1716.0    -166.0     4.6    -2.0    2265.0
  8    1345    -2063.0    -273.0     6.4    -2.0    2495.9
  9    1353    -2262.0    -139.0     5.9    -2.0    2200.0
 10    1360    -4920.0   -3832.0    11.3    -4.0    5163.6
 11    1365    -4207.0    -970.0    11.4    -3.0    5765.0
 12    1378    -7324.0   -2447.0    16.2    -2.0    9380.0
 13    1380    -2676.0    -598.0     5.8    -2.0    3675.0
 14    1480    -4024.0    -317.0    18.0    -4.0    5120.0
 15    1481    -3637.0    -883.0    18.0    -4.0    5112.0
 16    1546    -2176.0    -536.0     3.3    -2.0    2400.0
 17    1552    -1530.0   -1456.0     4.1    -2.0    2260.0
 18    1561    -1985.0   -1521.0     4.8    -2.0    2325.0
 19    1572    -4895.0    -314.0    11.5    -4.0    6210.0
 20    1573    -1831.0    -290.0     4.6    -2.0    2010.0
 21    1574    -5382.0   -5466.0    31.3    -2.0   14416.0
 22    1590    -2278.0    -256.0     5.6    -3.0    2192.0
 23    1599    -2963.0    -470.0     7.1    -2.0    3994.8
 24    1601    -4192.0    -589.0    10.7    -2.0    5640.0
 25    1616    -4598.0   -2285.0    13.0    -2.0    6400.0
 26    1620    -4049.0    -730.0    17.8    -3.0    5890.0
 27    1674    -7692.0    -134.0    25.8    -3.0   10391.0
 28    1759    -7274.0   -2378.0    37.1    -3.0   12037.8
 29    1808    -1855.0    -769.0     3.5    -2.0    1567.2
 30    1812    -1998.0   -1722.0     4.6    -1.0    2280.0
 31    1815    -2080.0   -1903.0    10.8    -2.0    2300.0
 32    1816    -2115.0   -1952.0     5.6    -2.0    2420.0
 33    1890    -7443.0    -626.0    22.7    -2.0   10127.0
 34    1915    -3903.0    -380.0     9.1    -2.0    4554.5
 35    1926    -3878.0    -317.0    10.5    -4.0    4506.0
 36    1927    -4901.0    -304.0    12.9    -3.0    6320.0
 37    1949    -3444.0   -1159.0     6.8    -2.0    3780.0
 38    1950    -6946.0   -4410.0    22.4    -3.0   10757.0
 39    1963    -8267.0    -717.0    19.2    -4.0   10030.0
 40    1976    -5056.0    -486.0    17.4    -4.0    5660.0
 41    1977    -5258.0    -768.0    15.5    -3.0    7554.7
 42    2002    -8737.0   -1067.0    21.7    -3.0    9820.0
 43    2006    -3572.0    -239.0     8.3    -2.0    5160.0
 44    2008    -6380.0    -624.0    18.0    -4.0    8090.0
 45    2015    -8479.0   -2949.0    17.3    -3.0    9729.0
 46    2017    -4453.0    -645.0     9.5    -3.0    5710.0
 47    2020    -6669.0   -1067.0    15.0    -4.0    7686.2
 48    2024    -4641.0    -463.0     9.9    -3.0    5163.0
 49    2031    -8824.0   -1474.0    18.6    -4.0    9700.0
 50    2184    -3043.0       0.0     6.9    -2.0    4004.0
```

```
ITERATIONS=   6

SOLUTION VALUE=     .371500F+04

     PRIMAL VARIABLES         DUAL SLACK                COST SENSITIVITY
  1        .100000F+01        0.                   -.964082E+03           .100000E+21

     DUAL VARIABLES         PRIMAL SLACK              RESOURCE SENSITIVITY
  1      -.166145F+01         0.                   -.100000E+21      0.
  2      0.                   0.                   0.                     .100000E+21
  3      -.584911E+00         0.                   -.100000E+21      0.
  4      -.182437F+01         0.                   -.100000E+21      0.

            1      1002             1.000

ITERATIONS=   9

SOLUTION VALUE=     .367731E+04

     PRIMAL VARIABLES         DUAL SLACK                COST SENSITIVITY
 10        .356080F+00        0.                   -.722502E+02           .209446E+04
 29        .750044E+00        0.                   -.672424E+03           .100188E+03
 31        .288343E+00        0.                   -.601341E+03           .419511E+02

     DUAL VARIABLES         PRIMAL SLACK              RESOURCE SENSITIVITY
  1      -.633102E+00         0.                   -.572205E+03           .796629E+03
  2      -.525145E+00         0.                   -.673237E+03           .330179E+03
  3      0.                    .286444E+06         -.286444E+06           .100000E+21
  4      -.704540E-02         0.                   -.171690E+04           .365159E+04

            2      1052             3.247

ITERATIONS=   7

SOLUTION VALUE=     .375293E+04

     PRIMAL VARIABLES         DUAL SLACK                COST SENSITIVITY
 10        .363403E+00        0.                   -.100693E+04           .212791E+03
 29        .119734E+01        0.                   -.313440E+04           .820369E+02

     DUAL VARIABLES         PRIMAL SLACK              RESOURCE SENSITIVITY
  1      -.936127E+00         0.                   -.100000E+21           .215814E+04
  2      0.                    .124531E+04         -.124531E+04           .100000E+21
  3      0.                    .765550E+05         -.765550E+05           .100000E+21
  4      -.108036E+00         0.                   -.420749E+04           .338701E+04

            3      1057             1.237
```

Example 6: A cost surface run:

CARD 1:   1, 4, 1, 0, 0, 1, 0, 6

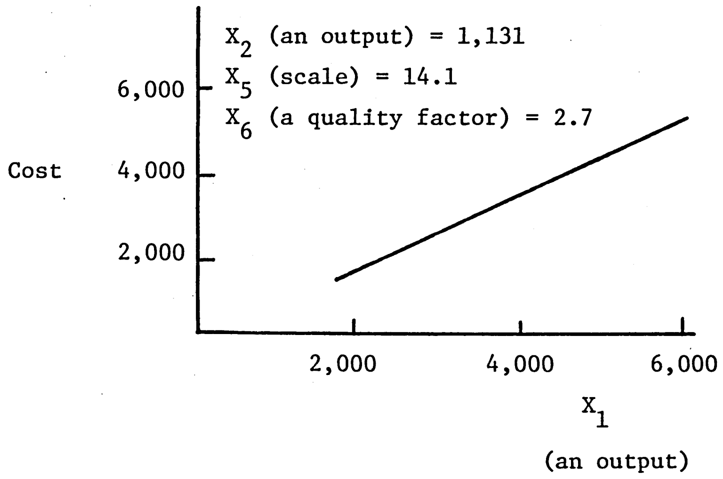
CARD 2:   -1, -2, 105, -106, 3, 4,

CARD 3:   20.0, 10.0

CARD(s) 4: DATA

CARD 6:   1, -2000.0, -4000.0, -6000.0

The output of this run is shown on the following pages. The data listing for this run is identical to the listing for example 5. The resulting portion of the frontier cost surface is illustrated below.



```
CONTROL PARAMETERS
          NUMBER OF REPLICATIONS =    1
          NUMBER OF CONSTRAINT VARIABLES =    4
          NUMBER OF EQUALITY CONSTRAINTS =    0
          MAXIMUM NUMBER OF INPUT VARIABLES FOR LEAST-COST ALGORITHM =6

VARIABLF INDEX LIST...         -1   -2  105 -106    3
THIS IS AN ISOQUANT RUN

THIS IS A LEAST-INPUT-COST RUN

VARIABLE MEANS ARE NOT TO BE RFAD FOR EACH REPLICATION

INPUT PRICES FOR COST ANALYSIS
          20.00      10.00
NUMBER OF OBSERVATIONS=    50

VARIABLE MEANS...
        -4327.3   -1130.8       14.1       -2.7      278.2       15.9
```

```
ISO-RUN VARIABLE INDICATOR AND VALUES
             1           -2000.     -4000.     -6000.
ITERATIONS=  11

SOLUTION VALUE=     .185662E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .133037E+00         0.                 -.722502E+02          .198539E+04
 29       .657227E+00         0.                 -.663057E+03          .100188E+03
 31       .607231E-01         0.                 -.584911E+03          .419511E+02

     DUAL VARIABLES         PRIMAL SLACK            RESOURCE SENSITIVITY
  1      -.631945E+00         0.                 -.127484E+03          .689584E+03
  2      -.524185E+00         0.                 -.595043E+03          .677578E+02
  3      0.                    .132644E+05       -.132644E+05          .100000E+21
  4      -.703252E-02         0.                 -.362232E+03          .136678E+04
ITERATIONS=  13

SOLUTION VALUE=     .364220E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .260983E+00         0.                 -.100693E+04          .205789E+03
 29       .146413E+01         0.                 -.248762E+04          .802694E+02

     DUAL VARIABLES         PRIMAL SLACK            RESOURCE SENSITIVITY
  1      -.910550E+00         0.                 -.100000E+21          .187252E+04
  2      0.                    .995244E+03       -.995244E+03          .100000E+21
  3      0.                    .280172E+05       -.280172E+05          .100000E+21
  4      -.105085E+00         0.                 -.528954E+04          .250076E+04
ITERATIONS=  10

SOLUTION VALUE=     .546330E+04

     PRIMAL VARIABLES         DUAL SLACK              COST SENSITIVITY
 10       .391475E+00         0.                 -.100693E+04          .205789E+03
 29       .219620E+01         0.                 -.248762E+04          .802694E+02

     DUAL VARIABLES         PRIMAL SLACK            RESOURCE SENSITIVITY
  1      -.910550E+00         0.                 -.100000E+21          .387252E+04
  2      0.                    .205825E+04       -.205825E+04          .100000E+21
  3      0.                    .420258E+05       -.420258E+05          .100000E+21
  4      -.105085E+00         0.                 -.793431E+04          .375114E+04
```

The Computer Program

The computer program for this algorithm is listed on the following pages. The program is written in FØRTRAN and it should be compatible with most computer systems. The author has run the program on a Burroughs 6700, an IBM 360/165, and a CDC 6400. A card deck for the program along with the sample data and control cards for the six illustrative examples are available upon request from the author.

```
      COMMON V(51,200),K(400),I(200),J(200),TS(200),F1(50),
     1F(400),PYM,ZL,H,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2IT,IS,ITC,NSI,NL,KAME,INVT
      DIMENSION INDEX(20),VMEAN(20),IDENT(200),VST(21,200),
     1VONE(10),VMST(20),COST(20)
      READ(5,10) NREP,NVAR,ISO,IEF,NEQ,ICST,IVM,ICN
10    FORMAT(8I5)
      KVAR=NVAR+1
      READ(5,20) (INDEX(II),II=1,20)
   20 FORMAT(20I4)
      ICX = ICN - KVAR + 1
      IF(ICST.EQ.1) READ(5,30) (COST(II),II=1,ICX)
   30 FORMAT(8F10.2)
      WRITE(6,40) NREP,NVAR,NEQ,ICN,
     1(INDEX(II),II=1,KVAR)
   40 FORMAT(1H1,'CONTROL PARAMETERS',/,10X,'NUMBER OF ',
     1'REPLICATIONS =',I4,/,10X,'NUMBER OF CONSTRAINT ',
     2'VARIABLES =',I4,/,10X,'NUMBER OF EQUALITY ',
     3'CONSTRAINTS =',I4,/,10X,'MAXIMUM NUMBER OF INPUT VARIABLES ',
     4'FOR LEAST-COST ALGORITHM =',I4,/,/,1X,'VARIABLE ',
     5'INDEX LIST...',5X,20I5,/)
      IF(ISO.EQ.0)  WRITE(6,50)
      IF(ISO.EQ.1)  WRITE(6,60)
      IF(ICST.EQ.0)  WRITE(6,70)
      IF(ICST.EQ.1)  WRITE(6,80)
      IF(IVM.EQ.0)  WRITE(6,90)
      IF(IVM.EQ.1)  WRITE(6,100)
   50 FORMAT(1X,'THIS IS NOT AN ISOQUANT RUN',/)
   60 FORMAT(1X,'THIS IS AN ISOQUANT RUN',/)
   70 FORMAT(1X,'THIS IS NOT A LEAST-INPUT-COST RUN',/)
   80 FORMAT(1X,'THIS IS A LEAST-INPUT-COST RUN',/)
   90 FORMAT(1X,'VARIABLE MEANS ARE NOT TO BE READ FOR EACH ',
     1'REPLICATION',/)
  100 FORMAT(1X,'VARIABLE MEANS ARE TO BE READ FOR EACH REPLICATION',/)
      IF(ICST.EQ.1) WRITE(6,110) (COST(II),II=1,ICX)
  110 FORMAT(1X,'INPUT PRICES FOR COST ANALYSIS',/,
     15X,10F10.2,/,5X,10F10.2,/)
      DO 120 II=1,21
      DO 120 JJ=1,200
  120 V(II,JJ)=0.0
      DO 130 II=1,20
  130 VMEAN(II)=0.0
      IF(IEF.EQ.1) KVAR = KVAR - 1
      NN=0
      IF(ICST.EQ.1) KVAR=ICN
  170 NN=NN+1
      READ (5,140)  IDENT(NN),(VST(II,NN),II=1,6)
  140 FORMAT (I6,4X,2F10.0,/,10X,2F10.2,/,60X,F10.6,/,30X,F10.0,/)
      IF(IDENT(NN).GT.999998) GO TO 150
      DO 160 II=1,KVAR
      KK=IABS(INDEX(II))
      IF(KK.GT.100) KK=KK-100
      SGN=1.0
      IF(INDEX(II).LT.0) SGN=-1.0
      V(II,NN)=SGN*VST(KK,NN)
      VMEAN(II)=VMEAN(II)+V(II,NN)
  160 CONTINUE
      GO TO 170
  150 KNT=NN-1
      WRITE(6,180) KNT
  180 FORMAT(1X,'NUMBER OF OBSERVATIONS=',I6,/)
      DO 190 II=1,KVAR
  190 VMEAN(II)=VMEAN(II)/FLOAT(KNT)
      WRITE(6,200) (VMEAN(II),II=1,KVAR)
  200 FORMAT(1X,'VARIABLE MEANS...',/,5X,10F10.1,/,
     15X,10F10.1,/)
      IF(IFF.EQ.1) KVAR = KVAR + 1
      IF(ICST.EQ.1) KVAR=NVAR+1
      M=KVAR
Y     IF(ICST.EQ.0) GO TO 210
      DO 220 II=1,KNT
      HOLD=0.0
      KK=0
      DO 230 JJ=KVAR,ICN
      KK=KK+1
  230 HOLD = HOLD + (COST(KK)*V(JJ,II))
      V(KVAR,II)=HOLD
  220 CONTINUE
  210 N=KNT+1
      M2=NEQ
      N2=0
      NL=0
      KAME=0
      IDP=0
      INVT=0
      NST=0
      ZL=1.0E-8
      WRITE(6,240)
  240 FORMAT(1H1,'LISTING OF DATA MATRIX',/,/)
      DO 250 II=1,KNT
      IF(IFF.EQ.1)V(M,II)=-1.0
      WRITE(6,260) II,IDENT(II),(V(JJ,II),JJ=1,KVAR)
  260 FORMAT(1X,I3,I8,2X,10F10.1,/,14X,10F10.1,/)
  250 CONTINUE
      DO 270 II=1,M
      DO 270 JJ=1,N
  270 VST(II,JJ)=V(II,JJ)
      DO 280 II=1,KVAR
  280 VMST(II)=999.9
      DO 290 NR=1,NREP
      KNTA=KNT
      IF(ISO.EQ.0) GO TO 3000 %**CHANGED FROM 300 TO 3000
      KNTA=3
      IF(IVM.EQ.1) READ(5,310) (VMST(II),II=1,KVAR)
  310 FORMAT(10F8.0)
      IF(IVM.EQ.1) WRITE(6,315) (VMST(II),II=1,KVAR)
  315 FORMAT(1H0,'VARIABLE MEANS FOR THIS REPLICATION',/,10(5X,10F10.1,/
     $))
      READ(5,320) IFRST,(VONE(II),II=1,3)
  320 FORMAT(I5,5X,3F10.0)
      WRITE(6,330) IFRST,(VONE(II),II=1,3)
  330 FORMAT(1H0,'ISO-RUN VARIABLE INDICATOR AND VALUES',/,
     110X,I5,5X,10F10.0,/)
      DO 340 II=1,KVAR
      IF(VMST(II).NE.999.9) VMEAN(II)=VMST(II)
  340 CONTINUE
 3000 DO 350 NA=1,KNTA      %** CHANGED FROM 300 TO 3000
      IF(ISO.EQ.0) GO TO 360
      VHOLD=VMEAN(IFRST)
      VMEAN(IFRST)=VONE(NA)
```

```
360 DO 370 II=1,M
    DO 370 JJ=1,N
370 V(II,JJ)=VST(II,JJ)
    DO 380 II=1,NVAR
    V(II,N)=VST(II,NA)
    IF(ISO.EQ.1) V(II,N)=VMEAN(II)
    KK=IABS(INDEX(II))
    IF(KK.LT.100) GO TO 380
    V(II,N)=0.0
    DO 390 JJ=1,KNT
    XA=VST(II,JJ)*VST(KVAR,JJ)
    XB=VMEAN(II)*VST(KVAR,JJ)
    IF(ISO.EQ.0) XB=VST(II,NA)*VST(KVAR,JJ)
    SGN=1.0
    IF(INDEX(II).LT.0) SGN=-1.0
    V(II,JJ)=SGN*(ABS(XA)-ABS(XB))
390 CONTINUE
380 CONTINUE
    V(KVAR,N)=0.0
    CALL LINEAR
    IF(ISO.EQ.1) VMEAN(IFRST)=VHOLD
    IF(ISO.EQ.1) GO TO 350
    EFF=999.0
    VL=V(M,N)
    IF(VL.NE.0.0) EFF=ABS(VST(KVAR,NA))/ABS(VL)
    WRITE(6,400) NA,IDENT(NA),EFF
400 FORMAT(/,2X,2I10,F15.3,/,/)
350 CONTINUE
290 CONTINUE
    STOP
    END


    SUBROUTINE LINEAR
    COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
   1F(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
   2LT,LS,ITC,NSI,NL,KAME,INVT
C     CONTROL PROGRAM AND INDEX SELECTION
  4 CALL READIN
    GO TO 13
C     EXECUTE PIVOT TRANSFORMATION
  8 ITC=ITC+1
    IF(ABS(V(LT,LS)).GT.ZL)GO TO 9
801 V(LT,LS)=0.0
    NO=NO+1
    GO TO 13
  9 CALL TRANS
C     INTERMEDIATE TABLEAU PRINT OUT
    IF(IDP.EQ.1)CALL PRINT
C     INDEX SELECTION
C     TAKE CARE OF ZL WIPEOUT
 13 IF(NO.LT.2)GO TO 12
    KC=NO-(2*(NO/2))
    L1=K(NO-KC)
    L2=K(NO-1+KC)
    IF(V(L1,L2).EQ.0.)NO=NO-1
 12 IF (NO.GT.2) GO TO 47
    IF (NO.EQ.2) GO TO 27
C     NO IS ONE
```

```
   15 NO = 2
      IDX1=N1+1
      IDX2= M - 1
      DO 21 IR=IDX1,IDX2
   18 IF (V(IR,N).LT.0.) GO TO 24
      IF (V(IR,N).EQ.0.) GO TO 21
   19 IF (IR.LE.LH) GO TO 21
   20 IF (I(IR).LE.M2) GO TO 24
   21 CONTINUE
      K(2) = M
      GO TO 27
   24 K(2) = IR
C     NO IS TWO
   27 IR = K(2)
      IDX1 = M1 + 1
      IDX2 = N - 1
      BST=0
      DO 39 IC=IDX1,IDX2
   31 IF (V(IR,IC).GT.0.) GO TO 34
      IF (V(IR,IC).EQ.0.) GO TO 39
   32 IF (IR.EQ.M) GO TO 44
   33 IF (V(IR,N).GE.0.) GO TO 36
      GO TO  44
   34 IF (IR.EQ.M) GO TO 35
   35 IF (V(IR,N).GT.0.) GO TO 44
   36 IF (IC.LE.LH) GO TO 39
   37 IF (J(IC).GT.N2) GO TO 39
   44 IF(ABS(V(IR,IC)).LE.BST)GO TO 39
   45 BST=ABS(V(IR,IC))
      K(3)=IC
   39 CONTINUE
      IF(BST.GT.0.)GO TO 46
C     FINAL PRINT OUT
      IF (K(2).NE.M) GO TO 42
      GO TO  43
   41 FORMAT('0 INCONSISTENT CONSTRAINTS'/(20I6))
   42 WRITE (6,41) I(IR),(J(IT),IT=1,LH)
   43 IDP=0
      CALL PRINT
      RETURN
   46 NO = 3
C     NO AT LEAST THREE
   47 KC = NO - (2 * (NO / 2))
      IF (KC.EQ.0) GO TO 121
C     NO ODD
C     SET SCANNING SEQUENCE
   51 IDX1 = N1 + 1
      IDX2 = NO - 4
      IDX3 = M - 1
      IF (IDX2.GT.0) GO TO 59
   55 NS = IDX3 - N1
      DO 57 IR=1,NS
   57 IS(IR) = IR + N1
      GO TO 69
   59 NS = 0
      DO 67 IR=IDX1,IDX3
      DO 64 IC=1,IDX2,2
      IDXCOL = K(IC)
      IF (V(IR,IDXCOL).NE.0.) GO TO 67
   64 CONTINUE
```

```
      NS = NS + 1
      IS(NS) = IR
   67 CONTINUE
C     DETERMINE TRANSFORMATION
   69 LS = K(NO)
      L1 = K(NO-1)
      L2 = K(NO-2)
      IF (L1.NE.M) GO TO 77
   73 EXTREM=1.0E20
      IF (V(L1,LS).LE.0.) GO TO 78
   75 EXTREM = -EXTREM
      GO TO 78
   77 EXTREM = V(L1,L2) / V(L1,LS)
   78 LT = L1
      DO 109IR=1,NS
      IDXR = IS(IR)
      IF (V(IDXR,LS).EQ.0.) GO TO 109
   82 RATIO = V(IDXR,L2) / V(IDXR,LS)
C     DECISION NET
      IF (RATIO.LT.0.) GO TO 105
      IF (RATIO.GT.0.) GO TO 102
   85 IF (LS.LE.LH) GO TO 97
   86 IF (J(LS).LE.N2) GO TO 90
   87 IF (IDXR.LE.LH) GO TO 94
   88 IF (I(IDXR).GT.M2) GO TO 94
C     SET TRANSFORMATION
   90 LT = IDXR
      GO TO 117
C     TEST FOR DEGENERACY
   94 IF (V(IDXR,LS).LE.0.) GO TO 109
   97 NO = NO + 1
      K(NO) = IDXR
      GO TO 13
C     TEST FOR EXTREME
C     RATIO POSITIVE
  102 IF (EXTREM.LE.0.) GO TO 109
  103 IF (EXTREM.LE.RATIO) GO TO 109
      GO TO  107
C     RATIO NEGATIVE
  105 IF (EXTREM.GE.0.) GO TO 109
  106 IF (RATIO.LE.EXTREM) GO TO 109
  107 LT = IDXR
  108 EXTREM  = RATIO
  109 CONTINUE
C     UNBOUNDED TABLEAU PRINT OUT
      IF (LT.NE.M) GO TO 115
      GO TO  112
  111 FORMAT('0EXTREME UNBOUNDED'I4)
  112 WRITE (6,111) J(LS)
      GO TO 43
  115 IF (LT.NE.L1) GO TO 117
  116 NO = NO - 1
  117 NO = NO - 1
      GO TO 8
C     NO EVEN
C     SET SCANNING SEQUENCE
  121 IDX1 = M1 + 1
      IDX2 = NO - 4
      IDX3 = N - 1
      IF (IDX2.GT.0) GO TO 129
  125 NS = IDX3 - M1
      DO 127 IC=1,NS
  127 IS(IC) = IC + M1
      GO TO 139
      DO 137 IC=IDX1,IDX3
      DO 134 IR=2,IDX2,2
      IDXR = K(IR)
      IF (V(IDXR,IC).NE.0.) GO TO 137
  134 CONTINUE
      NS = NS + 1
      IS(NS) = IC
  137 CONTINUE
C     DETERMINE TRANSFORMATION
  139 LT = K(NO)
      L1 = K(NO-1)
      L2 = K(NO-2)
      EXTREM  = V(L2,L1) / V(LT,L1)
      LS = L1
      DO 171 IC=1,NS
      IDXCOL = IS(IC)
      IF (V(LT,IDXCOL).EQ.0.) GO TO 171
  147 RATIO = V(L2,IDXCOL) / V(LT,IDXCOL)
C     DECISION NET
  129 NS = 0
      IF (RATIO.LT.0.) GO TO 167
      IF (RATIO.GT.0.) GO TO 164
  150 IF (IDXCOL.LE.LH) GO TO 156
  151 IF (J(IDXCOL).GT.N2) GO TO 156
C     SET TRANSFORMATION
  153 LS = IDXCOL
      GO TO 174
C     TEST FOR DEGENERACY
  156 IF (V(LT,IDXCOL).GE.0.) GO TO 171
  159 NO = NO + 1
      K(NO) = IDXCOL
      GO TO 13
C     TEST FOR EXTREME
C     RATIO POSTIVE
  164 IF (EXTREM.LE.0.) GO TO 171
  165 IF (EXTREM.LE.RATIO) GO TO 171
      GO TO  169
C     RATIO NEGATIVE
  167 IF (EXTREM.GE.0.) GO TO 171
  168 IF (RATIO.LE.EXTREM) GO TO 171
  169 LS = IDXCOL
      EXTREM = RATIO
  171 CONTINUE
      IF (LS.NE.L1) GO TO 174
  173 NO = NO -1
  174 NO=NO-1
      GO TO 8
      END


      SUBROUTINE TRANS
      COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
     1F(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2LT,LS,ITC,NSI,NL,KAME,INVT
C THIS TRANSFORMATION SUBROUTINE MODIFIED TO ZERO NEAR-ZERO ELEMENTS
```

```
C     FIRST STAGE
      ZT=ZL*.01
      DIV=V(LT,LS)
      V(LT,LS)=1.0
      NS=0
      DO 9 IC=1,N
      V(LT,IC)=V(LT,IC)/DIV
      IF(ABS(V(LT,IC)).GT.ZT)GO TO 8
      V(LT,IC)=0.
      GO TO 9
    8 NS=NS+1
      IS(NS)=IC
    9 CONTINUE
      DO 15 IR=1,M
      IF (IR.EQ.LT) GO TO 15
   11 X=V(IR,LS)
      IF (X.EQ.0.) GO TO 15
   12 V(IR,LS)=0.
      DO 14 IP=1,NS
      IC=IS(IP)
      V(IR,IC)=V(IR,IC)-X*V(LT,IC)
      IF(ABS(V(IR,IC)).LE.ZT)V(IR,IC)=0.
   14 CONTINUE
   15 CONTINUE
      IF (LT.LE.LH) GO TO 18
   17 IF (LS.LE.LH) GO TO 20
      GO TO  40
   18 IF (LS.LE.LH) GO TO 53
      GO TO  28
C     ROW INTERCHANGE
   20 LTEMP = I(LS)
      I(LS) = I(LT)
      I(LT) = LTEMP
      IF (I(LS).GT.M2) GO TO 26
   24 CALL INCHC (LS,M1+1,1)
      M1 = M1 + 1
   26 RETURN
C     COLUMN INTERCHANGE
   28 LTEMP = J(LS)
      J(LS) = J(LT)
      J(LT) = LTEMP
      IF (J(LT).GT.N2) GO TO 39
   36 CALL INCHR(LT,N1+1,1)
      N1 = N1+1
   39 RETURN
C     ADD ROW AND COLUMN
   40 CALL INCHR(LT,LH+1,0)
      CALL INCHC(LS,LH+1,0)
      IF (I(LH+1).GT.M2) GO TO 47
   45 CALL INCHC (LH + 1, M1 + 1,1)
      M1 = M1 + 1
   47 IF (J(LH+1).GT.N2) GO TO 50
   48 CALL INCHR (LH + 1,N1 + 1, 1)
      N1 = N1 + 1
   50 LH = LH + 1
      RETURN
C     DELETE ROW AND COLUMN
   53 CALL INCHR(LT,LH,1)
      CALL INCHC(LS,LH,1)
      LH = LH - 1
      RFTURN
      FND


      SUBROUTINE INCHR(LR1,LR2,LR3)
      COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
     1E(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2LT,LS,ITC,NSI,NL,KAME,INVT
      DO 7 IC=1,N
      TFMP = V(LR1,IC)
      V(LR1,IC) = V(LR2,IC)
    7 V(LR2,IC) = TFMP
      IF (LR3.GT.0) GO TO 11
    8 LTEMP = I(LR1)
      I(LR1) = I(LR2)
      I(LR2) = LTEMP
      GO TO 12
   11 LTEMP = J(LR1)
      J(LR1) = J(LR2)
      J(LR2) = LTEMP
   12 DO 13 IR=2,NO,2
      IF (LR2.EQ.K(IR)) GO TO 15
   13 CONTINUE
      GO TO 16
   15 K(IR) = LR1
   16 RETURN
      FND


      SUBROUTINE INCHC(LC1,LC2,LC3)
      COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
     1F(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2LT,LS,ITC,NSI,NL,KAMF,INVT
      DO 7 IR=1,M
      TFMP = V(IR,LC1)
      V(IR,LC1) = V(IR,LC2)
    7 V(IR,LC2) = TFMP
      IF (LC3.GT.0) GO TO 11
    8 LTEMP = J(LC1)
      J(LC1) = J(LC2)
      J(LC2) = LTEMP
      GO TO 12
   11 LTEMP = I(LC1)
      I(LC1) = I(LC2)
      I(LC2) = LTEMP
   12 DO 13 IC=1,NO,2
      IF (LC2.EQ.K(IC)) GO TO 15
   13 CONTINUE
      GO TO 16
   15 K(IC) =LC1
   16 RETURN
      FND


      SUBROUTINE READIN
      COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
     1F(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2IT,LS,ITC,NSI,NL,KAME,INVT
      ZI=1.0D- 8
      IF(KAME .EQ.0.) GO TO 19
      DO 14 IR=1,M
   14 V(IR,N+1)=1.0
      DO 15 IC=1,N
   15 V(M+1,IC)=-1.0
C     TRANSLATE TO POSITIVE PAYOFF
      PYM=0.0
      DO 16 IR=1,M
      DO 16 IC=1,N
      IF(V(IR,IC) .GE. PYM) GO TO 16
      PYM=V(IR,IC)
   16 CONTINUE
      IF(PYM .GE. 0.0) GO TO 18
      DO 17 IR=1,M
      DO 17 IC=1,N
   17 V(IR,IC)=V(IR,IC) -PYM
   18 M=M+1
      N=N+1
      V(M,N) =0.
   19 IF(INVT .EQ. 0) GO TO 33
C     TRANSPOSE TO DUAL
      LTEMP=M
      M = N
      N = LTEMP
      LTEMP = M2
      M2 = N2
      N2 = LTEMP
      IF (M.GE.N) GO TO 22
   20 IDX1=N
      GO TO 23
   22 IDX1=M
   23 DO 27 IR=1,IDX1
      DO 27 IC=IR,IDX1
      TEMP=-V(IR,IC)
      V(IR,IC)=-V(IC,IR)
   27 V(IC,IR)=TEMP
      DO 29 IC=1,N
   29 V(M,IC) = -V(M,IC)
      DO 31 IR=1,M
   31 V(IR,N) = -V(IR,N)
C     COMPLETE STEP SETUP
   33 LH= 0
      M1 = 0
      N1 = 0
      NS = 0
      LT = 0
      LS = 0
      ITC=0
      IS (1) =0
      DO 34 IR=1,M
   34 I(IR)=IR
      DO 36 IC=1,N
   36 J(IC)=IC
      NO=1
      K(1)=N
C     INITIAL TABLEAU PRINT OUT
      IF(IDP .EQ. 1) CALL PRINT
   42 RETURN
```

```
      FND


      SUBROUTINF PRINT
      COMMON V(21,200),K(400),I(200),J(200),IS(200),F1(20),
     1F(400),PYM,ZL,M,M1,M2,N,N1,N2,LH,IDP,IX2,NO,NS,
     2LT,LS,ITC,NSI,NL,KAME,INVT
      DIMENSION VT(300),SL(300),RL(300),RU(300)
      DATA AEQ,ATQ,AFV,ANV,RHS,WW/2HEC,2HIC,2HFV,2HNV,2HRS,2HBR/
      IF(IDP.FQ.0)GO TO 36
C     TABLEAU PRINT OUT
      IF(ITC.EQ.0)GO TO 10
    4 FORMAT(7H PIVOT(I3,1H,I3,1H))
      WRITE(6,4)LT,LS
    9 FORMAT('0' //'  ITERATION=',I4)
   10 WRITF(6,9)TTC
   11 FORMAT(21H EQUATIONS IN KERNEL=I3)
      WRITE(6,11)M1
   13 FORMAT(26H FREE VARIABLES IN KERNEL=I3)
      WRITF(6,13)N1
   15 FORMAT(25H CURRENT CONTROL SEQUENCE)
      WRTTF(6,15)
   17 FORMAT(7H K(NO)=28I4/(1H 30I4))
      WRITE(6,17)(K(IR),IR=1,NO)
   19 FORMAT(21H CUPRENT KERNEL SIZE=I3)
      WRTTF(6,19)LH
   21 FORMAT(21H BASIC TABLEAU V(M,N))
      WRITE(6,21)
      DO 22 JC=1,N
      IF (JC.LE.LH)IS(JC)=I(JC)
      IF(JC.GT.LH)IS(JC)=J(JC)
      IF((JC.LE.LH).AND.(JC.LE.M1))VT(JC)=AEQ
      IF((JC.LE.LH).AND.(JC.GT.M1))VT(JC)=AIQ
      IF((JC.GT.LH).AND.(J(JC).LE.N2))VT(JC)=AFV
   22 IF((JC.GT.LH).AND.(J(JC).GT.N2))VT(JC)=ANV
      VT(N)=RHS
   24 FORMAT (8(I13,A2))
      WRITE(6,24)(IS(IC),VT(IC),IC=1,N)
   26 FORMAT(1H ,I3,A2,8E15.6/(6H      ,8E15.6))
      DO 28 IR=1,M
      IF(IR.LF.LH)IS(IR)=J(IR)
      IF(IR.GT.LH)IS(IR)=I(IR)
      IF((IR.LE.LH).AND.(IR.LE.N1))VT(IR)=AFV
      IF((IR.LE.LH).AND.(IR.GT.N1))VT(IR)=ANV
      IF((IR.GT.LH).AND.(I(IR).LF.M2))VT(IR)=AFQ
      IF((IR.GT.LH).AND.(I(IR).GT.M2))VT(IR)=AIQ
      IF(IR.EQ.M)VT(IR)=WW
   28 WRITF(6,26)IS(IR),VT(IR),(V(IR,IC),IC=1,N)
      RETURN
C     FINAL PRINT OUT
   35 FORMAT(' ITERATIONS=',I4)
   36 IF(KAME.GE.1)GO TO 118
      WRITE(6,35) ITC
   44 FORMAT(16H0SOLUTION VALUF=F15.6)
      VL=-V(M,N)
      WRITE(6,44)VL
   47 FORMAT(1H0,5X,16HPRIMAL VARIABLES,7X,10HDUAL SLACK,12X,16HCOST SEN
     1SITIVITY)
      WRITF(6,47)
```

```
C     SET PRINT VECTORS
      NF=N-1
      DO 58 JC=1,NF
      JLM=J(JC)
      IF(JC.GT.LH)GO TO 56
      VT(JLM)=V(JC,N)
      SL(JLM)=0.0
      GO TO 58
   56 SL(JLM)=V(M,JC)
      VT(JLM)=0.0
   58 CONTINUE
      M3=M1+1
      DO 76 J1=1,NF
      LLM=J(J1)
      IF(J1.GT.LH)GO TO 74
      RGU=1.0E20
      RGL=-1.0E20
      DO 70 J2= M3,NF
      IF(V(J1,J2).EQ.0.)GO TO 70
      RATIO= V(M,J2)/V(J1,J2)
      IF(RATIO.LT.0.)GO TO 68
      IF(RATIO.GE.RGU)GO TO 70
      RGU=RATIO
      GO TO 70
   68 IF(RATIO.LE.RGL)GO TO 70
      RGL=RATIO
   70 CONTINUE
      RL(LLM)=RGL
      RU(LLM)=RGU
      GO TO 76
   74 RL(LLM)=-V(M,J1)
      RU(LLM)=1.0E20
   76 CONTINUE
   77 FORMAT(1H I3,E17.6,E20.6,E18.6,E18.6)
      DO 79 JC=1,NF
      IF(VT(JC).EQ.0.0) GO TO 79
      WRITE(6,77)(JC,VT(JC),SL(JC),RL(JC),RU(JC))
  79  CONTINUE
C     SET DUAL PRINT VECTORS
   81 FORMAT(1H05X,14HDUAL VARIABLES,7X,12HPRIMAL SLACK,10X,20HRESOURCE
     1SENSITIVITY)
      WRITE(6,81)
      MF=M-1
      DO 92 IC=1,MF
      MLM=I(IC)
      IF(IC.GT.LH)GO TO 90
      VT(MLM)=-V(M,IC)
      SL(MLM)=0.0
      GO TO 92
   90 SL(MLM)=V(IC,N)
      VT(MLM)=0.0
   92 CONTINUE
      N3 = N1 + 1
      DO 112 I1=1,MF
      NLM=I(I1)
      IF(I1.GT.LH)GO TO 110
      RGU=1.0E20
      RGL=-1.0E20
      DO 106 I2=N3,MF
      IF(V(I2,I1).EQ.0.)GO TO 106
      RATIO=-V(I2,N)/V(I2,I1)
      IF(RATIO.LT.0.)GO TO 104
      IF(RATIO.GE.RGU)GO TO 106
      RGU=RATIO
      GO TO 106
  104 IF(RATIO.LE.RGL)GO TO 106
      RGL=RATIO
  106 CONTINUE
      RL(NLM)=RGL
      RU(NLM)=RGU
      GO TO 112
  110 RL(NLM)=-V(I1,N)
      RU(NLM)=1.0E20
  112 CONTINUE
      DO 114 IC=1,MF
  114 WRITE(6,77)(IC,VT(IC),SL(IC),RL(IC),RU(IC))
      RETURN
C     GAME THEORY OUTPUT
  117 FORMAT(12H0GAME VALUE=F15.6)
  118 GV=( 1.0/V(M,N))+PYM
      WRITE(6,117)GV
      MF=M-1
      DO 122 IC=1,MF
  122 VT(IC)=0.0
      IF(LH.EQ.0)GO TO 127
      DO 125 IC=1,LH
      MLM=I(IC)
  125 VT(MLM)=V(M,IC)/V(M,N)
  126 FORMAT(11H0ROW PLAYER)
  127 WRITE(6,126)
  128 FORMAT(I3,F11.6)
      WRITE(6,128)((IC,VT(IC),IC=1,MF))
      NF=N-1
      DO 132 JC=1,NF
  132 VT(JC)=0.0
      IF(LH.EQ.0)GO TO 137
      DO 135 JC=1,LH
      JLM=J(JC)
  135 VT(JLM)=V(JC,N)/V(M,N)
  136 FORMAT(14H0COLUMN PLAYER)
  137 WRITE(6,136)
      WRITE(6,128)((JC,VT(JC)),JC=1,NF)
  139 RETURN
      END
```

REFERENCES

[1] Aigner, D. and S. Chu, "On Estimating the Industry Production Function," American Economic Review, September 1968, pp. 826-837.

[2] Boles, J., "Efficiency Squared . . . Efficient Computation of Efficiency Indexes", Western Farm Econ. Assoc., 1966 Proceedings, Pullman, Washington, January 1967, pp. 137-142.

[3] _________, "The 1130 Farrell Efficiency System - Multiple Products, Multiple Factors," Giannini Foundation of Agricultural Economics, February 1971.

[4] _________, "The Measurement of Productive Efficiency: The Farrell Approach," U.C. Berkeley, unpublished manuscript, 1972.

[5] Carlson, D., "The Production and Cost Behavior of Higher Education Institutions," Paper P-36, Ford Foundation Program for Research in University Administration, December 1972.

[6] ___________, "Examining Efficient Joint Production Processes", Measuring and Increasing Academic Productivity, Robert Wallhaus (ed.), Jossey-Bass, San Francisco, Winter 1975.

[7] Danø, S., Industrial Production Models, Springer-Verlag, New York, 1966.

[8] Farrell, M., "The Measurement of Productive Efficiency," Journal of the Royal Statistical Society, Series A, Part III, 1957, pp. 253-290.

[9] Farrell, M. and M. Fieldhouse, "Estimating Efficiency Production Functions Under Increasing Returns to Scale," Journal of the Royal Statistical Society, Series A, Part II, 1962, pp. 252-267.

[10] Henderson, J. and R. Quandt, Microeconomic Theory: A Mathematical Approach, Second Edition, McGraw-Hill, 1971.

[11] Seitz, W., "Productive Efficiency in the Steam-Electric Generating Industry," Journal of Political Economy, July/August 1971, pp. 878-886.

[12] Sitorus, B., "Productive Efficiency and Redundant Factors of Production in Traditional Agriculture of Underdeveloped Countries," Proceedings, 39th Annual Meeting, Western Farm Economics Association, 1966.

[13] Timmer, C., "Using a Probabilistic Frontier Production Function to Measure Technical Efficiency," Journal of Political Economy, July/August 1971, pp. 776-794.