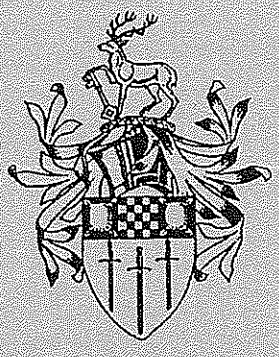
T P
SEEDS Technical Paper

SURREY
ENERGY
ECONOMICS
CENTRE

Using LAMBDA for DEA

David Hawdon and Ian M McQueen

April 1996



Technical Paper No.1
ISBN 1852371811

Department of Economics
University of Surrey

Surrey Energy Economics Centre Technical Paper No.1

Using LAMBDA for DEA

David Hawdon, University of Surrey
and
Ian M McQueen, University of Reading

ISBN 1852371811, April 1996

1 Introduction to DEA

Data Envelopment Analysis is a nonparametric technique for fitting an efficient production frontier using data on inputs and outputs from a sample of decision making units (DMUs). It is particularly well suited to the analysis of multiple output organisations where ordinary production function estimation techniques may not be applicable. It can also handle situations where outputs and inputs may not be defined in terms of ordinary economic variables, as in the case of public sector institutions such as hospitals, government departments and public utilities.

The method has some interesting characteristics:-

1. It is based on an optimising procedure rather than a statistical one. This means that it can be applied very naturally to problems in economics and management where the objective is to maximise outputs or to minimise inputs. On the other hand it does not directly address the problem of uncertainty in data which usually affect observations of economic activity.

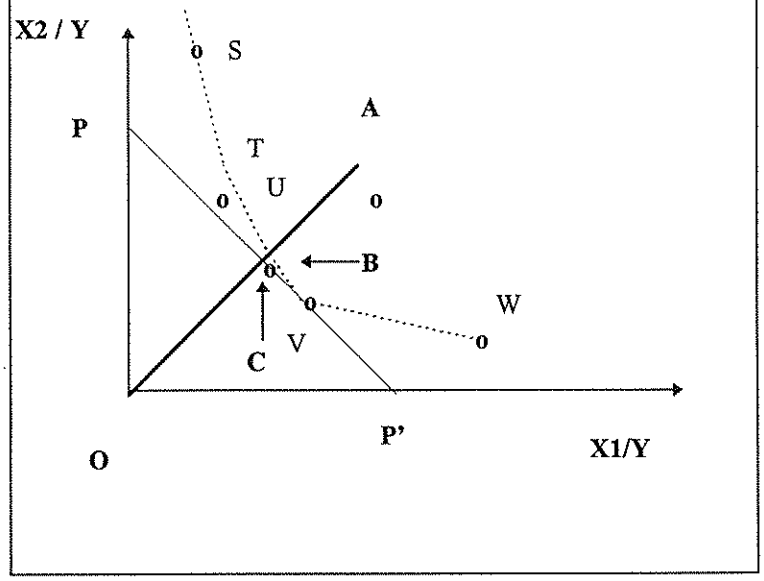
2. It does not require pre-specification of a functional form for the production frontier. This means that it does not incur the misspecification errors commonly associated with econometric investigations. It does not enable parameters for a given frontier to be estimated although rates of substitution can be calculated from the efficient frontier.

3. It identifies the best reference set for each decision making unit and is therefore of use to management. Where the DMU is not efficient, those efficient DMUs nearest in structure to the inefficient DMU are indicated and weights are calculated which can be used as guidance for improvement.

2. Theory of Data Envelopment Analysis

DEA is clearly explained by Seiford and Thrall (1990) in a collection of papers on productivity estimation in the Journal of Econometrics. The relationship of DEA to production frontiers is fully developed in Fare et al (1994) which contains many interesting suggestions for further applications of the technique. Consider a set of single output organisations using two inputs, X1 and X2 to produce Y. Divide each input by its corresponding output, to get the amount of input required to support a unit level of output. Figure 1 shows various combinations of inputs which support a unit level of output, marked o.

Figure 1: Inputs, Output and The Efficient Frontier



The efficient frontier is drawn through points representing DMUs S,T,U,V,W which use minimum amounts of inputs X1 and X2 to produce unit levels of output. DMU A, however, although efficient in its use of X2, uses a larger amount of X1 than is necessary and lies to the left of the efficient frontier. Firms like A are technically inefficient in that the same amount of output could be produced for less input by adopting the technology of firms on the frontier. Firms lying on the frontier are not technically inferior to other firms and may be taken to represent 'best practice'. They form a reference set against which the performance of other firms may be judged.

The technical inefficiency of a firm A can be measured by the ratio of a line drawn from the origin to a point B on the frontier divided by length OA. The ratio will either be 1, if the firm is fully efficient, or less than 1 if the firm is inefficient. Where input prices are known, it is also possible to find the economic efficiency of the DMU. Assume that the input price line is PP'. Economic efficiency is calculated by the ratio of the line from the origin to the point where it crosses the factor price line, OC, to the entire line OA. Economic efficiency will of course be at least equal to technical efficiency.

Mathematically the data envelopment problem is to find for each firm, a set of weights λj on the technologies of all the firms in the reference set, which place the firm as nearly as possible to the efficient frontier. Clearly if the firm is efficient, the solution will involve only its own technology. If it is not

efficient, the calculated weights will supply information on resource minimising ways of improving efficiency. This result can be achieved by solving for each firm the following linear program:

$$\text{Min } h_o = \theta$$

Subject to

$$\sum_{j=1} X_{ij}\lambda_j \leq \theta_o X_{io} \qquad (1)$$

$$\sum_{j=1} Y_{rj}\lambda_j \geq Y_{ro}$$

$$\lambda \geq 0$$

where each firm (j = 1,....,n) consumes varying amounts of m inputs (X_{ij}, i = 1,....,m) to produce s different outputs (Y_{rj}, r = 1,....,s). For any value of θ , the first constraint says that the actual inputs used by firm j, X_{io} are at least as great as they would be by adopting some other firms technology out of the reference set. The second constraint states that the actual amount of output produced by firm j could have been more than exceeded by adopting the input combinations of some other firm(s). Minimising (1) has the effect of finding the technology which most nearly places firm j onto the efficient frontier. The weights on the technologies adjust during the programming solution to assist this process. The end result is a solution which is most favourable to each of the separate firms since it does not impose weights on the constraints but allows them to emerge from the optimisation procedure. Where improvement of performance is possible, a unique formula for such improvement is obtained for each firm.

The problem as it stands yields a solution under the assumption of constant returns to scale. Useful information on the existence of economies of scale can be found by adding the following constraints to the problem:

$\sum \lambda = 1$ to yield variable returns to scale, or

$\sum \lambda \leq 1$ to yield increasing non increasing returns to scale.

Comparison of results can show whether there are ranges of increasing, constant and diminishing returns to scale. For a practical illustration see Bjurek, Hjalmarsson and Forsund, (1990).

3. The LAMBDA program

LAMBDA implements DEA by performing the following functions:

1. It converts data input into a form suitable for use with a standard linear programming (LP) package,

2. It solves the program for each DMU in turn.

3. Finally, output is directed to a file which can be inspected and edited with any word-processing package.

The package used is the freely available LP_SOLVE package, written by M.R.C.M. Berkelaar of Eindhoven University of Technology*. LP-SOLVE is a powerful package in its own right, reputed to be capable of handling up to 10,000 variables. Designed for use with any PC., it can handle maximisation and minimisation problems involving =, ≤, and ≥ constraints for integer as well as linear programming problems. Although intuitive in its data input procedures, and therefore very convenient in comparison with commercial LP packages, the specifications are not suitable for the DEA user. LP_SOLVE requires for instance that every constraint should be fully written out with variables, coefficients and operators at each stage. Typical DEA problems involve many variables (one for each DMU) and it is preferable to avoid repetition of variable names for entry. What LAMBDA does is to make data input relatively straightforward. Data should be stored in a file as follows with each line ended by ;

S;	The number of outputs, in this case 2.
min:;	Objective (alternative is max:)
1;	Always 1
$X_{11}\ X_{12}\ X_{13}\ X_{14} X_{15} X_{16} \ldots\ldots\ X_{1N} \leq 0$;	Data on first input, by firm, separated by 1 blank
$X_{21}\ X_{22}\ X_{23}\ X_{24}\ X_{25} X_{26} \ldots\ldots\ X_{2N} \leq 0$;	Data on second input by firm
.	
$X_{M1}\ X_{M2}\ X_{M3}\ X_{M4}\ X_{M5}\ X_{M6} \ldots\ldots\ X_{MN} \leq 0$;	Data on Mth input by firm
$Y_{11}\ Y_{12}\ Y_{13}\ Y_{14} Y_{15}\ Y_{16} \ldots\ldots\ Y_{1N} \geq 0$;	Data on first output
.	
$Y_{S1}\ Y_{S2}\ Y_{S3}\ Y_{S4}\ Y_{S5} Y_{S6} \ldots\ldots\ Y_{SN} \geq 0$;	Data on Sth output
[EOF];	

Always end each row with ; and the final row with only one carriage return.
The data should be saved to a DOS file.

Running LAMBDA

Type **LAMBDA**

Enter the name of the **data file** when requested. LAMBDA will convert your data entry into a form suitable for processing by LP_SOLVE, and will echo back a copy to the screen.

LAMBDA asks for the **name of a file to which output will be sent**. After processing is complete, this file should contain:

Data i, i.e. results for firm i

Value of objective function: θ

Weights (λj) attached to each firm (xj) in reference set.

4. Example of a two input, two output model with 6 firms:

Suppose you have data on 2 inputs and 2 outputs for 6 DMUs as follows:-

	DMU 1	DMU 1	DMU 1	DMU 1	DMU 1	DMU 1
Input 1	18.8	34.5	25	0.4	0.3	1.2
Input 2	3.8	16.6	1.9	0.1	0.02	0.02
Output 1	13.4	52.2	5.9	0.4	0.08	0.005
Output 2	4	3.6	3.5	4.9	3.9	0.4

Invoke the Editor and create the following file:

2;

min:;

1;

18.8 34.5 25 0.4 0.3 1.2 <= 0;

3.8 16.6 1.9 0.1 0.02 0.02 <= 0;

13.4 52.2 5.9 0.4 0.08 0.005 >= 0;

4 3.6 3.5 4.9 3.9 0.4 >= 0;

[EOF];

Save it as '**elec**

Type '**LAMBDA**'

Input file? **elec**

Elec.dat is then printed on the screen.

min:

1x7;

18.8x1+ 34.5x2+ 25x3+ 0.4x4+ 0.3x5+ 1.2x6- 1.2x7 <= 0;

3.8x1+ 16.6x2+ 1.9x3+ 0.1x4+ 0.02x5+ 0.02x6- 0.02x7 <= 0;

13.4x1+ 52.2x2+ 5.9x3+ 0.4x4+ 0.08x5+ 0.005x6 >= 0.005;

4x1+ 3.6x2+ 3.5x3+ 4.9x4+ 3.9x5+ 0.4x6 >= 0.4;

Output file? **elec.out**

The results are in file elec.out.

Data 1

Value of objective function:	0.8815789584
x1	0
x2	0
x3	0
x4	30.615
x5	14.426
x6	0
x7	0.88158

Data 2

Value of objective function:	0.9999999854
x1	0
x2	1
x3	0
x4	0
x5	0
x6	0
x7	1

Data 3

Value of objective function:	0.7763157992
x1	0
x2	0
x3	0
x4	14.75
x5	0
x6	0
x7	0.77632

Data 4

Value of objective function:	0.9999999805
x1	0
x2	0
x3	0
x4	1
x5	0
x6	0
x7	1

Data 5

Value of objective function:	0.9999999755
x1	0
x2	0
x3	0
x4	0
x5	1
x6	0
x7	1

Data 6

Value of objective function: 0.1025641001

x1	0
x2	0
x3	0
x4	0
x5	0.10256
x6	0
x7	0.10256

The results imply that of the six decision making units, three (DMUSs 2, 4 and 5) were on the efficient frontier. Of the others, DMU 1 was 88% efficient, DMU 3 was 77% and DMU 6 was only 10% efficient in comparison with the effiicient frontier. DMU 1's efficiency could be improved most effectively by imitating the technologies of DMUs 4 and 5 using the weights 30.6 and 14.4. As the following calculation shows, the weighted combination of 4 and 5s technology would produce the same amount of output 1 (13.4 units) for less of inputs 1 and 2, while yielding a greatly increased output of good 2. Social efficiency would be increased by substituting the weighted combination of DMUs 4 and 5 technology for that presently used by DMU 2.

	DMU 1	DMU 4	DMU 5	30.6 x DMU 4	14.26 x DMU 5	30.6DMU4 + 14.26DMU 5
Input 1	18.8	0.4	0.3	12.2	4.3	16.5
Input 2	3.8	0.1	0.02	3	0.3	3.3
Output 1	13.4*	0.4	0.1	12.3	1.2	13.4*
Output 2	4	4.9	3.9	169	56.3	219.3

Comparison of efficient with actual inputs confirms that inefficiency here is measured in input terms. Note that the two weights do not add up to 1 because the problem is solved in terms of original units of output and input.

5. How LAMBDA Works

The program expects the data file to contain required characters. Each line of data must end with the semicolon character, and the last line of the datafile must end with [EOF]. The purpose of the semicolon at the end of each line allows long lines of data to be read in, each data line can be split simply by not ending the line with a semicolon and can be split any number of times. The [EOF] enables the program to know where the relevant data ends, thus further comments may be added after this marker.

The operation of the program consists of three distinct phases. First the data file is read through and the number of lines containing the data are read. This allows a dynamic array to be initialised ahead of the second phase thus reducing memory requirements of the program.

The second phase again reads the data but this time a line of data is read, ignoring carriage returns and line feeds and is then parsed. A line is regarded as everything up to a semicolon or end of file marker. The program looks for the first non-space character of the line and assumes this is the first numeric character. Each character is then read until the next space or looks for the first non-space character of the line and assumes this is the first numeric character. Each character is then read until the next space or end of line is found, this group of characters are then stored in the appropriate location of the two parameter array. This continues for the whole length of the line and the whole file is read and stored in this way resulting in a matrix array of the data; thus each value can be referenced simply by its row and column position in the array.

The third phase first asks for the output file for results to be stored. It then simply loops through the matrix array, row by row, adding onto each value the appropriate "Xn +" characters and then adding the required component at the end of each line. This is written to a temporary file and the program calls the "LP_SOLVE" program with this temporary file and the output file as part of its parameters. This process loops through until all values of the equation are used.

6. Requirements and Limitations

Although the software works on any PC with DOS 5 or above, a fast processor is recommended.

There is no practical limit to the number of inputs and outputs.

7. Availability of software

Copies of LAMBDA can be obtained from

David Hawdon,
Department of Economics,
University of Surrey,
Guildford,
Surrey,
GU2 5XH.
E-mail:d.hawdon@surrey.ac.uk

LAMBDA is issued subject to the restriction that it must only be used for academic purposes.

Information on LP_SOLVE can be obtained from:

* M.R.C.M. Berkelaar
Eindhoven University of Technology
Design Automation Section
P.O. Box 513
NL-5600 MB Eindhoven, The Netherlands
phone ...-31-40-473345
E-mail: michel@es.ele.tue.nl

8. References

Bjurek, H., Hjalmarsson, L., Forsund, F.R. (1990)*Deterministic Parametric and Nonparametric estimation of Efficiency in service Production*. Scandinavian Journal of Economics, 213-217.

Fare, R., Grosskopf, S., and Lovell, C. A. K. (1994) *Production Frontiers*. Cambridge University Press.

Seiford, L.M. and Thrall, R.M. (1990) *Recent developments in DEA*. Journal of Econometrics, 46, 39-56.

APPENDIX

DATA PREPARATION USING EXCEL AND WORD

Data can be entered into EXCEL for graphical and statistical analysis and then converted into a format suitable for LAMBDA. The procedure is as follows:

1. After entering data in EXCEL, transpose if necessary so that data on each variable is in rows.

2. Copy the data to the clipboard

3. Minimise EXCEL and activate WORD

4. Paste Special the data into WORD

5. Use unformatted data option of Paste Special

6. Note that each piece of data is separated by a TAB.

7. Highlight the data, Edit Replace the TABs with single spaces.

8. Add constraints, ;, and [EOF] to the data set.

9 Save in DOS format for work with LAMBDA.

INFORMATION ABOUT SURREY ENERGY ECONOMICS CENTRE (SEEC)

SEEC consists of members of the Department of Economics who work on energy and environmental economics. SEEC's aims include the promotion of research and teaching in the broad area of energy economics and policy. SEEC was founded in 1983. Its creation was intended to consolidate the research on energy economics which developed at the University of Surrey after the appointment of Colin Robinson to the Chair of Economics in 1968. Colin and the colleagues he attracted to the University built up Surrey's reputation, initially with pioneering work on North Sea oil and gas, and subsequently in all the other major areas of energy economics.

- Recent research covers the following areas: Electricity, gas and coal privatisation in the UK; privatisation in the Middle East; the structure of UK energy demand; North Sea oil and gas economics and policy: international oil markets; electricity economics; the transport of energy; environmental policy in the UK; energy and environmental policy in the Third World; global environmental issues.

- SEEC research output includes **SEEDS** - Surrey Energy Economics Discussion paper Series - as well as a range of other academic papers, books and monographs, including ***SEEC Occasional Papers series*** and **TP** - SEEDS Technical Papers.

- Each year SEEC organises a range of energy conferences and workshops on energy themes. Specialist workshops include the meetings of the joint SEEC/BIEE (British Institute of Energy Economics) **Energy Modelling Group**, convened by David Hawdon and Paul Appleby (B.P).

- Members of SEEC provide major inputs into the postgraduate energy courses run by the Economics Department - in particular, the M.Sc. courses on Energy Economics and Energy Policy, and the Diploma in the Economics of Energy and Development for Graduates.

Enquiries to:

David Hawdon
Director of SEEC and Editor of
E-mail: D.Hawdon@surrey.ac.uk

Secretary: Isobel Hildyard
E-mail: I.Hildyard@surrey.ac.uk

SEEC, Economics Department, University of Surrey, Guildford, GU2 5XH, UK.
Tel: +44-(0)1483 259379 FAX: +44-(0)1483 303775