An Extension to SAS/OR® for Decision System Support
Ali Emrouznejad
Higher Education Funding Council for England,
Northavon house, Coldharbour Lane,
Bristol, BS16 1QD
U.K.

ABSTRACT
This paper explores the use of the optimisation procedures in SAS/OR software with application to the decision system support using Data Envelopment Analysis (DEA) techniques. Data Envelopment Analysis is a linear programming method for assessing the efficiency and productivity of Decision Making Units. Over the last decade DEA has gained considerable attention as a managerial tool for measuring performance of organisations and it has been widely used for assessing the efficiency of public and private sectors such as banks, airlines, hospitals, universities and manufactures. As a result, new applications with more variables and more complicated models are being introduced.

On the other hand, the SAS® System has numerous optimisation routines that handle the standard optimisation problems such as linear programming with all types of constraints. To facilitate the use of DEA by SAS users, a SAS/DEA code was implemented in the SAS programming language. The SAS macro developed in this paper selects the chosen variables from a SAS data file and constructs a Linear-programming model based on the selected DEA. An example is given to illustrate how can one use the code to measure the efficiency of organisations.

KEYWORDS:
SAS, Data Envelopment Analysis, Linear Programming

INTRODUCTION
Data Envelopment Analysis (DEA) is a linear programming method for assessing the efficiency and productivity of Decision Making Units. DEA continues to grow in importance as managerial tools become more reliable and handle performance measurement of organisations. As a result, new applications with more variables and more complicated models are being introduced. On the other hand, the SAS System has numerous optimisation routines that handle the standard optimisation problems such as linear programming with all types of constraints.

The main purpose of this paper is to implement a SAS/DEA System. This is followed by an illustration of how the user could use this system.

The paper unfolds as follows. Next section describes the procedures of the SAS System and SAS/OR products. In section 2 the DEA models and method of calculations in DEA are explained. The remainder of this paper illustrates features that have been added to the SAS System for the purpose of DEA modeling, call it SAS/DEA. These include procedures for data handling, model building and reporting the results of DEA model. Finally, this paper uses a small data set to show some of the features of the SAS/DEA system.

ABOUT SAS
Today SAS is recognised as a one of the lead packages for statistical analysis and as a powerful tool for data base systems in many organisations, both in public and private sectors. SAS users come from every major industry (banking to pharmaceuticals, manufacturing to telecommunications, ...) all with the same basic needs to make better strategic decisions and to gain competitive edge.

There are many applications in SAS that the users recognised as powerful tools in organisational management. For example the SAS/OR System has numerous optimisation procedures which handle the standard problems such as linear and non-linear programming with all types of constraints. These capabilities are exposed to the SAS users in a variety of places such as in procedures LP and NLP3.

In addition to the standard procedures available in SAS System applications such as neural network, simulation4 and control project management5 are introduced. The absence of Data Envelopment Analysis code in SAS force the users to use the other packages while the SAS/OR is very powerful and has capability of handling all types of DEA models. Therefore this paper aims to introduce a new application in SAS System for performance measurement of Decision Making Units using Data Envelopment Analysis (DEA) technique.

ABOUT DEA
Decision Making Units (DMUs) are units of organisations such as banks, universities, and hospitals, which typically perform the same function. A DMU usually uses a set of inputs (resources) to secure a set of outputs (products). DEA is a method for measuring efficiency of DMUs using linear programming techniques to “envelop” observed input - output vectors as tightly as possible. One main advantage of DEA is that it allows several inputs and several outputs to be considered at the same time. In this case, efficiency is measured in terms of inputs or outputs along a ray from the origin.

Assume a set of observed DMUs, \([DMU_j ]; j=1,\ldots,n]\), associated with \(m\) inputs, \([x_{ij}; i=1,\ldots,m]\), and \(s\) outputs, \([y_{ij}; r=1,\ldots,s]\). In the method originally proposed by Charnes, Cooper, Rhodes (1978) the efficiency of the DMU \(j_0\) is defined as follows.

Model 1. Output oriented - CRS model

\[
\begin{align*}
\text{Max } & h \\
\text{s.t. } & \sum \lambda_i x_{ij} + S_i^+ = x_{ij} \quad \forall i \\
& \sum \lambda_j y_{oj} - S_r^- = hy_{ij} \quad \forall r \\
& S_i^+, S_r^- \geq 0 \quad \forall i, \forall r \\
& \lambda_j \geq 0 \quad \forall j.
\end{align*}
\]

Where;
- \(x_{ij}\) = the amount of the \(i^{th}\) input at DMU \(j\),
- \(y_{ij}\) = the amount of the \(r^{th}\) output from DMU \(j\) and
- \(j_0\) = the DMU to be assessed.

If \(h^*\) is the optimum value of \(h\), then DMU \(j_0\) is said to be Pareto...
efficient if \( h^* = 1 \) and the optimal values of \( S_i^- \) & \( S_r^+ \) are zero for all \( i \) & \( r \). This means no other DMU or combination of DMUs exist which can produce at least the same amount of output as DMU \( j_0 \), with less for some resources and/or more for any other resources.

In Model (1), \( S_i \) and \( S_r \) represent slack variables. Thus a slack in an input \( i \), i.e. \( S_i > 0 \), represents an additional inefficiency use of input \( i \). A slack in an output \( r \), i.e. \( S_r^- > 0 \), represents an additional inefficiency in the production of output \( r \).

The DEA Model (1) is known as output - oriented model because it expands output of DMU \( j_0 \) within the production space. It should be solved \( n \) times once for each DMU being evaluated to generate \( n \) optimal values of \(( h^*, \lambda^* )\).

For DMU \( j_0 \)

If radial expansion is possible Model (1) will yield \( h^* > 1 \).

If radial expansion is not possible Model (1) will yield \( h^*_j = 1 \).

The positive elements of the optimal values in \( \lambda \) identify the set of dominating DMUs located on the constructed production frontier, against which the DMU \( j_0 \) is evaluated. This subset of DMUs is called "peers" to DMU \( j_0 \). From Model (1) it is clear that the model defines the relative efficiency of a DMU in terms of output maximisation. The input oriented model of DEA can be defined in a similar way.

**Model 2. Input oriented - CRS model**

\[
\begin{align*}
\text{Min } & \phi \\
\text{s.t.} & \\
\sum_{j} \lambda_j x_{ij} + S_{i}^- &= \phi x_{ij_0} \quad \forall i \\
\sum_{j} \lambda_j y_{rj} - S_{r}^- &= y_{rj_0} \quad \forall r \\
S_{i}^+, S_{r}^- &\geq 0 \quad \forall i, r \\
\lambda_j &\geq 0 \\
\end{align*}
\]

Where:

\( x_{ij} \) = the amount of the \( i \)th input at DMU \( j \),

\( y_{rj} \) = the amount of the \( r \)th output from DMU \( j \) and

\( j_0 \) = the DMU to be assessed.

Assume that \( \phi^* \) is the optimum value of \( \phi \). DMU \( j_0 \) is said to be Pareto efficient iff \( \phi^* = 1 \) and the optimal value of \( S_i^- \) and \( S_r^+ \) are zero \( \forall i, r \).

Besides developing DEA in theory, practitioners in a number of fields have quickly recognised that DEA is a useful methodology for measuring productivity and efficiency. Therefore a number of software was developed. Some of the well known packages which researchers frequently use is: Warwick DEA Software, Frontier Analysis, IDEAS and GAMAS DEA\(^7\). The first three are commercial and they allow some DEA analysis to be undertaken, there are only few options to change in details of the model used\(^8\). Warwick DEA software has ability to calculate the effect of weight restriction and target setting\(^9\). The advantage of GAMS DEA is that it has the capability of building other DEA models. However GAMS is not popular software in organisational management. Many of the organisations such as Banks, Universities, Hospitals, Telecommunications, Airlines and so on use SAS software as a data base system or statistical analysis. The SAS/DEA introduced in this paper provides a powerful management tool for assessing the efficiency of organisations in SAS system.

**SAS/DEA**

The SAS/ DEA software developed in this paper solves DEA models. The program can handle both input minimisation and output maximisation. To enhance model there are several parameters. The user can select the desired parameters according to the particular model that is required. Users familiar with SAS can add their own features to build other DEA models. Users not familiar with SAS need only to run the program with their model specification prior running the system.

The SAS/DEA requires two initial data sets that contain the input and output variables for observed units. The data describing inputs/outputs must be presented in the format that variables appear in columns and units in rows and saved as .txt file (Tab delimited). The program has the ability to accommodate unlimited number of inputs/outputs with unlimited number of DMUs. The only limitation is the memory of computer used to run the SAS/DEA.

The SAS/DEA software then converts data sets to a DEA model selected. Based on the data and parameters specified in the SAS/DEA, the code first create the usual linear program then it use "Proc LP" to solve the model. Results then will be transferred to report files.

The SAS/DEA produces a table of efficiencies of DMUs. It also supplies much other valuable information including lambdas in primal and weights in dual DEA models. These information are very useful for analysing the inefficient units, where the source of inefficiency comes from and how could improve an inefficient unit to the desired level.

Figure (1) illustrates the data flow in the SAS/DEA. It shows

- the input and output variables data sets that are required for SAS/DEA.
- the parameters that SAS/DEA needs to define the model.
- The macros; (%data, %model and %report) and
- the result data sets that are available for print or can be used for further analysis.

**Figure 1. Data flow in SAS/DEA**

In the rest of this paper the procedure of implementation of SAS/DEA together with an example are explained.

**DEFINITION OF TERMS AND TYPOGRAPHICAL CONVENTIONS**

In the rest of this paper and particularly in the SAS/DEA code we will see several type styles used. Style conventions are summarised below:

- Courier font is used to show example of SAS statements. In most cases, this paper uses lowercase type for SAS code. The user can enter own SAS code in lowercase, uppercase or a mixture of the two. Enter any titles and footnotes exactly as you want them to appear on the printout.

- Variable name that are surround by " _ " are specifically used as parameters to the SAS. In all case these variables must be used without any change.

- Variable name that are started with " _ " are specifically used as parameters to the SAS/DEA. In all case these variables can be redundant if default values are adopted.

The terms in Table 1 are used for defining parameters needed in SAS/DEA. All these parameters are arbitrary and in absence of them the default value will be assigned.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%data</td>
<td>Input variables data file</td>
</tr>
<tr>
<td>%model</td>
<td>Output variables data file</td>
</tr>
<tr>
<td>%report</td>
<td>Parameters for SAS/DEA. In all cases these variables can be used without any change.</td>
</tr>
</tbody>
</table>
Table 1. The terms used for defining parameters in SAS/DEA

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>_InpData</td>
<td>Specifies the file name of input variables</td>
</tr>
<tr>
<td>_NoInut</td>
<td>Specifies the number of the input variables in the optimisation DEA model</td>
</tr>
<tr>
<td>_OutData</td>
<td>Specifies the file name of output variables</td>
</tr>
<tr>
<td>_NoOutput</td>
<td>Specifies the number of the output variables in the optimisation DEA model</td>
</tr>
<tr>
<td>_Orienta</td>
<td>This parameter is used to specify the orientation of the DEA model to be solved. &quot;InputMin&quot; for input minimisation model and &quot;OutputMax&quot; for output maximisation model</td>
</tr>
</tbody>
</table>

The SAS/DEA runs three macros for data handling (_data_), model building (_model_) and report writing (_report_).

**DATA HANDLING (_data_)**

This part of SAS/DEA reformats the data to a suitable format that can be used in SAS/OR. The SAS/DEA requires two data sets including input and output variables. Both data sets need to be "*.txt" files saved as "text tab delimited". Variables must be presented in columns and units in rows. The unit names must start with a letter and may contain up to 50 characters and must be listed in the first column of the data file. The other columns are including numeric values of variables. These variables can be entered into the file in any order. The first row of the data file represents the name of the variables. An example of data file for input variables is:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Input1</th>
<th>Input2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMU1</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>DMU2</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>DMU3</td>
<td>15</td>
<td>40</td>
</tr>
</tbody>
</table>

The data file for output variable has similar format.

There are four parameters prior to calling the data macro: _InpData, _NoInput, _OutData, _NoOutput_.

The purpose of each parameter is presented in Table 1.

**MODEL BUILDING (_model_)**

This part of SAS/DEA builds the requested DEA model in the format suitable for SAS/OR and for purpose of calling "Proc LP". There is one parameter prior calling the procedures: _Orienta_.

For example, for solving a standard input minimisation model the user should set the parameters to:

- _Orienta=InputMin;

and for solving output maximisation model the user should set the parameters to:

- _Orienta=OutputMax;

SAS procedure for model building is as follows.

```sas
%macro model;
%local i jj 0;
%let _nUnit=&_nDmu;
array NYarray(&_nOutput) $ NY1-NY&_nOutput;
array AYarray(&_nOutput) AY1-AY&_nOutput;
array NXarray(&_nInput) $ NX1-NX&_nInput;
array AXarray(&_nInput) AX1-AX&_nInput;
array Jarray(&_nUnit) U1-U&_nUnit;
length _row_ $ 50 _col_ $ 50 _type_ 8; keep _row_ _col_ _type_ _coef_ ;
J0=&jj; /* Jo is unit under assessment*/
if _&_Orienta='InputMin' then do;
  _type_='MIN';
  _row_='OBJ';
  _col_ ';'
  _coef_=.;
  output;
  _type_='.';
  _row_='OBJ';
  _col_='FI';
  _coef_=1;
  output;
  _type_='.';
  _row_='_rhs_';
  _col_='';
  _coef_=0;
  output;
end;
link ReadNX;
do j= 1 to &_nUnit;
  call symput('_nDmu', _n_-1);
run;
%mend model;
```

`link ReadNX;`
do i= 1 to &_nInput;
  _col_='LA' || put(j,3.);
  _row_=NXarray(i);
  _type_='.';
  _coef_=AXarray(i);
  output;
if j=J0 then do;
  select(&_Orienta);
  when('InputMin') do;
    _col_='FI';
    _row_=NYarray(i);
    _type_='GE';
    _coef_=-AXarray(i);
    output;
    _col_='_rhs_';
    _row_=NYarray(i);
    _type_='LE';
    _coef_=0;
    output;
  end;
  when('OutputMax') do;
    _col_='FI';
    _row_=NYarray(i);
    _type_='GE';
    _coef_=AXarray(i);
    output;
    _col_='_rhs_';
    _row_=NYarray(i);
    _type_='LE';
    _coef_=0;
    output;
  end;
  otherwise;
  end;
end;
end;
end;
end;
ReadNX: set NX; return;
ReadAX: set AX; return;
run;

data MODEL2;
  %let _nUnit=&_nDmu;
  array NYarray(&_nOutput)$ NY1-NY&_nOutput ;
  array AYarray(&_nOutput) AY1-AY&_nOutput;
  array NXarray(&_nInput) $ NX1-NX&_nInput ;
  array AXarray(&_nInput) AX1-AX&_nInput;
  array Jarray(&_nUnit) U1-U&_nUnit ;
  length _row_ $ 50 _col_ $ 50 _type_$8;
  keep _row_ _col_ _type_ _coef_ ;
J0=&jj; * Jo is unit under assessment;
if &_Orienta='OutputMax' then do;
  _type_='MAX';
  _row_ ='OBJ';
  _col_='.';
  _coef_=_value_; output;
  _type_='.';
  _row_ ='OBJ';
  _col_='.';
  _coef_=1; output;
  _type_='.';
  _row_ ='OBj';
  _col_='.';
  _coef_=0; output;
end;
else do j=1 to &_nUnit;
  model;
  proc lp data=MODEL noprint sparsedata primalout=lp2 dualout=lp3 ; run;
  data EffJ0(keep=Uj0 eff); set lp2 (where=(_var_='OBJ'));
  Uj0=&jj;
  if &_Orienta='OutputMax' and _value_ne 0 then Eff=_value_; else Eff=1/_value_; run;
  proc datasets nolist; append base=Eff
  end;
  proc sort data=Eff; by eff; run;
  proc print data=Eff; run;
  proc sort data=Eff; by UnitName; run;
  proc print data=Eff; run;
%mend model ;

SAS/DEA REPORT WRITING (%REPORT)
The SAS/DEA results are including table of efficiencies of DMUs in two formats sorted by the most efficient DMU and sorted alphabetically by the name of DMUs. All other information like slacks and lambdas in model (1 or 2) are saved in file "Report1". All information on the dual to model (1 or 2) including weights are saved in file "Report2".

SAS procedure for report writing is as follows.
%macro report;
  data eff(drop=Uj0);
    merge DMUs Eff;
    by Uj0;
  run;
  proc sort data=Eff; by eff; run;
  proc print data=Eff; run;
%mend report;

SAS/DEA MACRO
To make the system as ease as possible the "%sasdea macro" put all the above code together.
%macro sasdea;
  libname sasdea 'C:sasdea';
  proc datasets nolist; delete Eff Report1 report2; run;
  %data;
  %let _nUnit=&_nDmu;
  %do jj=1 %to &_nUnit;
    %model;
    proc lp data=MODEL noprint sparsedata primalout=lp2 dualout=lp3 ; run;
    data EffJo(keep=Uj0 eff) ; set lp2 (where=(_var_='OBJ'));
    Uj0=&jj;
    if &_Orienta='OutputMax' and _value_ne 0 then Eff=1/_value_; else Eff=_value_; run;
    proc datasets nolist; append base=Eff
The above code manages to get the result based on the specified parameters and data files.

**SAMPLE RESULTS FROM SAS/DEA**

The result of running above code is presented below. The third column shows the efficiency score.

<table>
<thead>
<tr>
<th>OBS</th>
<th>UNITNAME</th>
<th>EFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1.00000</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>1.00000</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>1.00000</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>1.00000</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>0.97750</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>0.86745</td>
</tr>
</tbody>
</table>

As this table shows units E and F are inefficient units, all other units are efficient. This is exactly matched with the results reported by Sexton in Table 5 (page 22). Unit F is less efficient than unit E. The best that unit F can do is 0.867, while the maximum of unit E is 0.977.

SAS/DEA also creates a wealth of information and saved them in two files, report1 and report2. By using these in SAS system we can specify a management strategy indicating by how much each unit should reduce each input and/or increase each output in order to become perfectly efficient. To see this we need the value of slacks and lambdas in primal models (1 or 2) and value of weights in dual to each. These all are saved in files report1 and report2 respectively. It is important to use these reports effectively to derive the measure required for managerial purpose.

For example the lambdas’ value for unit E can be obtained from report1 by using fields _var_ and _value_. It can be seen that the value of lambdas for unit E are as follows:

<table>
<thead>
<tr>
<th>OBS</th>
<th><em>VAR</em></th>
<th><em>VALUE</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>LA A</td>
<td>0.20000</td>
</tr>
<tr>
<td>4</td>
<td>LA B</td>
<td>0.08048</td>
</tr>
<tr>
<td>5</td>
<td>LA C</td>
<td>0.00000</td>
</tr>
<tr>
<td>6</td>
<td>LA D</td>
<td>0.53833</td>
</tr>
<tr>
<td>7</td>
<td>LA E</td>
<td>0.00000</td>
</tr>
<tr>
<td>8</td>
<td>LA F</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

What these values suggest is that the target for unit E has an input output levels that are 0.2 times those for unit A plus 0.08048 times those for unit B plus 0.53833 those for unit D. Therefore the optimum value of input output variables for unit E are as follows:

<table>
<thead>
<tr>
<th>StHr</th>
<th>Supp</th>
<th>MCPD</th>
<th>PPPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>342</td>
<td>1.17</td>
<td>19000</td>
<td>25000</td>
</tr>
</tbody>
</table>

Thus unit E should try to reduce its input levels by 2.3% without reducing either output levels. (Note that, 342 equals 97.7% of 350 and 1.17 equals 97.7% of 1.2).

**CONCLUSION**

Today many organisations recognised SAS as a one of the lead packages for data base system and statistical analysis. In
particular, optimisation procedures in SAS/OR are exposed to the user in a variety of places such as "Proc LP" and "Proc NLP". Therefore many applications such as neural network and control project management are introduced. SAS/DEA as introduced in this paper is a new application in SAS/OR that is a powerful managerial tool for measuring the efficiency and productivity of Decision Making Units.

The SAS/DEA application has several advantages including:

- It can be used by SAS users with no extra cost to buy any DEA software.
- It is very powerful with no limitation on the input and output variables or number of DMUS. The only limitation is the memory and disk space of the computer uses.
- It is flexible to add other DEA model.
- A SAS/DEA for end user is as ease as to run one statement: `%sasdea`.
- And importantly is SAS has powerful managerial tools for report writing and statistical analysis as well as optimisation procedures. The SAS/DEA report's files can directly feed to other SAS routines for further analysis. The results can dynamically be used in conjunction with cost reduction or profit maximisation in decision system support.

REFERENCES


ACKNOWLEDGMENTS

The author wishes to express his thanks to John Thompson and Judy Akinbolu for their valuable comments and suggestions. Acknowledgement is also due to "Higher Education Funding Council for England" for their financial support.