XI Balkan Conference on Operational Research
BALCOR 2013
Belgrade, Zlatibor, 07-10 September 2013
UNIVERSITY OF BELGRADE
FACULTY OF ORGANISATIONAL SCIENCES

AN AMPL OPTIMIZATION SOFTWARE LIBRARY FOR DATA ENVELOPMENT ANALYSIS

GEORGE ARGYRIOY[1], ANGELO SIFALERAS[1]

[1] University of Macedonia, School of Information Sciences, Department of Applied Informatics, 156 Egnatia Str., 54006 Thessaloniki, Greece
Emails: {tm0829, sifalera}@uom.gr

Abstract: *This work presents a new optimization software library, which includes a number of Data Envelopment Analysis (DEA) models. All the DEA models have been implemented in the AMPL algebraic modeling language for mathematical programming and solved using either Gurobi Optimizer v5.5 or MINOS v5.51 (for those models having general nonlinear objectives). In order to verify the correctness of the results, real world datasets have been examined and solved using the well-known commercial DEA Solver Pro v8.0c by SAITECH, Inc. We also present a literature review of both commercial and non-commercial DEA software packages and discuss about the benefits and drawbacks of our proposed DEA model library.*

Keywords: *Optimization Software, AMPL, Data Envelopment Analysis*

1. INTRODUCTION

DEA constitutes a non-parametric approach based on linear optimization models and methods, in order to construct a non-parametric surface over the available data. Performance -or efficiency- measurement of each DMU can be computed, regarding its position on this surface. DEA evaluates the relative performance of Decision Making Units (DMUs) (e.g., universities, hospitals) and can easily handle multiple input and output of each DMU. The efficient DMUs define an efficient production possibility surface or frontier.

A large number of DEA models has been reported in the scientific literature during the last 30 years [6]. DEA models have been applied for the measurement of the relative performance in many situations, such as in the food production industry [17], e-health providers [18], and many more diverse types of DMUs. The majority of these DEA models belong into four main categories: i) radial, ii) non-radial and oriented, iii) non-radial and non-oriented, and iv) composite of radial and non-radial. The classification between radial and non-radial models depends whether we are interested in a proportionate change of input/output or not. The classification between oriented and non-oriented models depends whether we are mainly interested rather in input reduction or output expansion, or not (e.g., input reduction and output expansion at the same time).

The proposed DEA software package provides a number of DEA models. Apart from the basic DEA models either with constant returns-to-scale (CCR) or variable returns-to-scale (BCC) characteristics, our DEA model library features implementations of DEA models with increasing (IRS) or decreasing (DRS) returns-to-scale characteristics. There are also included other implementations of DEA models, e.g., to handle slacks-based measure (SBM) of efficiency. Moreover, the proposed DEA software package may also solve either radial or non-radial super efficiency models. Furthermore, the software package can solve either input or output oriented versions of the above DEA models.

Section 2 presents a literature review concerning either commercial or non-commercial DEA software packages as also the motivation of this research work. Following, in section 3 the mathematical formulations of all the DEA models that are currently implemented in our optimization software library are presented. It should be noted that, the formulations of these DEA models have the same notation as in [7]. Implementation details and an example of a DEA model in AMPL is given in section 4. Finally, in section 5 we discuss about some future research directions.

2. LITERATURE REVIEW AND MOTIVATION

2.1. Commercial DEA software packages

There are several commercial software packages for the solution of Data Envelopment Analysis problems. Apart from their price, all these DEA software packages mainly differ on the number of DEA implemented models, in the variety of the produced reports, etc. A list of some of the state-of-the-art commercial DEA software packages is briefly presented in this subsection. Although, this list is no extensive, it describes some of the most well known DEA applications. For a recent, more analytical, review of DEA software packages, both commercial and non-commercial, the readers can refer to the work by Barr in [4].

DEA Solver Pro by SAITECH Inc.[1] is a user-friendly, spreadsheet based, DEA software package which includes a large number of popular DEA models (e.g., dynamic DEA, network DEA). A detailed report of the features of the latest version of DEA Solver Pro can be found in [15]. Furthermore, KonSi DEA Analysis[2] is also a well-known DEA software package which handles up to 250 input and output variables and produces a large number of reports including efficiency scores table, histogram, graph, and others. KonSi DEA 3D Visualization is a tool that provides 3D visualization of efficient frontier for CRS and VRS models with one input and two output parameters.

PIM-DEASOFT[3] is a DEA software package developed under supervision of Dr. A. Emrouznejad and Prof. E. Thanassoulis, which has several useful features such as extensive data handling facilities for very large data sets, ability to assess productivity change over time, graphical output facilities, and also permits data importing and exporting to Excel. Banxia Frontier Analyst[4] provide several DEA models with “undesirable”, categorical and other types of variables, and also it allows user to undertake Malmquist productivity analysis. Moreover this application features a fully threaded calculation engine in order to be multi-processor capable and thus split an analysis into multiple parts.

DEAFrontier[5] developed by Prof. J. Zhu is a spreadsheet add-in for the solution of DEA models. This application includes, among others, many of the standard DEA approaches such as cross efficiency, bootstrapping, two-stage network, or DEA-based supply chain model. Roughly speaking, this application solves more than 150 different DEA models and it is described in [23]. Finally, DEA Online Software (DEAOS.com[6]) is web-based software provided by Behin-Cara Co. Ltd. This service is capable for the solution of either single periodic or multiple periodic models, for a maximum of 500 units and unlimited indices in every project.

2.2. Non-commercial DEA software packages

Apart from the previously mentioned commercial (and usually expensive) DEA applications, several research efforts have been also made in order to provide free, quality, DEA software packages. The majority of these DEA software packages is built using a general-purpose optimization modeling language (either open source or not). However, there is no extra cost in order to obtain these software packages by the respective developers. Such general-purpose optimization modeling languages provide user with flexibility to modify the models and easily develop new, more complicated DEA models. Furthermore, a user is also able to select an open source solver in conjunction with the modeling language.

FEAR[7] (Frontier Efficiency Analysis with R), is a free DEA software library which can be linked to the general-purpose statistical package R, and was developed by Prof. P. W. Wilson. In order to use FEAR, one should first download and install the R language. The set of included routines are highly flexible, thus allowing measurement of technical, allocative, and overall efficiency. Furthermore, there are publicly available versions for either Linux, Mac OS, or Windows systems. A short paper introducing FEAR can be found in [22].

The modeling abilities of the General Algebraic Modeling System (GAMS) have been already utilized by several authors, e.g., [14, 9, 12], in order to provide DEA software implementations[8]. Moreover, Green [11] presented an implementation of the Charnes-Cooper-Rhodes [5] based on a Modeling Language for Mathematical Programming (AMPL). Also, Emrouznejad had presented the implementation of DEA models using macros of the SAS programming language [8]. Kleine and Sebastian have also developed a free web-based DEA-tool[9]. This tool allows user to process various DEA models such as additive models, range adjusted models, super-efficiency models, Russell-Measure, and others [13]. Open Source DEA (OSDEA[10]) consists a free and open source DEA solver library that can be used in java programs. User can write their own models, by modifying the java code of the OSDEA Library.

---

[1] http://www.saitech-inc.com/Products/Prod-DSP.asp
[2] http://www.dea-analysis.com
[3] http://www.deasoftware.co.uk
[4] http://www.banxia.com/frontier
[5] http://www.deafrontier.net/software.html
[6] https://www.deaos.com
[7] http://www.clemson.edu/economics/faculty/wilson/Software/FEAR/fear.html
[8] http://www.gams.com/modlib/libhtml/dea.htm
[9] http://www.dea.uni-hohenheim.de
[10] http://www.opensourcedea.org

2.3. Motivation

Although there is a plethora of successful commercial DEA software packages, we believe that more research efforts are still required in order to develop more complete non-commercial software packages. Thus, our goal was to develop a library with a large number of state-of-the-art DEA models. Since the proposed DEA software library is built using a general-purpose optimization modeling language such as AMPL, researchers will find it convenient that with little effort, modifications can be easily made and provide new DEA models. The benefits of the proposed DEA software library are first that it utilizes the rich features of the AMPL modeling language, second that the code of the DEA models can be easily extended, and third that the majority of the Operational Research (OR) scientists are more familiar to general-purpose optimization modeling languages, rather than other programming languages. Finally, the variety of the DEA models that are already implemented also consists a strong point of the proposed DEA software package.

3. DEA MODELS

3.1. Input-oriented Charnes-Cooper-Rhodes model

This is the input-oriented version (dual problem) of the most basic DEA model, presented in [5]:

$$\begin{aligned} \min_{\theta,\lambda} z &= \theta \\ s.t.\ \theta x_o - X\lambda &\geq 0 \\ Y\lambda &\geq y_o \\ \lambda &\geq 0 \end{aligned} \qquad (CCR-I)$$

The efficiency score is denoted by a scalar θ.

3.2. Ouput-oriented Charnes-Cooper-Rhodes model

The output-oriented version, is formulated as follows:

$$\begin{aligned} \max_{\eta,\mu} z &= \eta \\ s.t.\quad x_o - X\mu &\geq 0 \\ \eta y_o - Y\mu &\leq 0 \\ \mu &\geq 0 \end{aligned} \qquad (CCR-O)$$

3.3. Input-oriented Banker-Charnes-Cooper model

The input-oriented version of the classic DEA model, presented in [2], differs from the CCR-I model only in the addition of the convexity condition (e is the unity vector):

$$\begin{aligned} \min_{\theta,\lambda} z &= \theta \\ s.t.\ \theta x_o - X\lambda &\geq 0 \\ Y\lambda &\geq y_o \\ e\lambda &= 1 \\ \lambda &\geq 0 \end{aligned} \qquad (BCC-I)$$

3.4. Ouput-oriented Banker-Charnes-Cooper model

The output-oriented version, is formulated as follows:

$$\begin{aligned} \max_{\eta,\lambda} z &= \eta \\ s.t.\qquad X\lambda &\leq x_o \\ \eta y_o - Y\lambda &\leq 0 \\ e\lambda &= 1 \\ \lambda &\geq 0 \end{aligned} \qquad (BCC-O)$$

3.5. Input-oriented Slacks-based measure model

The formulation of the input-oriented Slacks-based measure model, presented in [19], is as follows:

$$\begin{aligned} &\min_{\lambda, s^-} z = 1 - \frac{1}{m} \sum_{i=1}^{m} \frac{s_i^-}{x_{io}} \\ &s.t.\ x_o = X\lambda + s_i^- \\ &\quad y_o \leq Y\lambda \\ &\quad \lambda, s^- \geq 0 \end{aligned} \qquad (SBM-I-C)$$

3.6. Ouput-oriented Slacks-based measure model

The formulation of the output-oriented Slacks-based measure model, is as follows:

$$\begin{aligned} &\min_{\lambda, s^+} z = \frac{1}{1 + \frac{1}{s} \sum_{r=1}^{s} \frac{s_r^+}{y_{ro}}} \\ &s.t.\ x_o \geq X\lambda \\ &\quad y_o = Y\lambda - s^+ \\ &\quad \lambda, s^+ \geq 0 \end{aligned} \qquad (SBM-O-C)$$

3.7. Free Disposal Hull model

The Free Disposal Hull (FDH) model described in [21], aims to evaluate the DMUs based only on their actually observed input and output values. A mixed-integer programming formulation of the FDH model follows:

$$\begin{aligned} &\min_{\theta, \lambda} z = \theta \\ &s.t.\ \theta x_o - X\lambda \geq 0 \\ &\quad y_o - Y\lambda \leq 0 \\ &\quad e\lambda = 1 \\ &\quad \lambda_j \in \{0, 1\} \end{aligned} \qquad (FDH)$$

3.8. Input-oriented increasing returns-to-scale model

The formulation of the input-oriented increasing returns-to-scale (IRS) model, described in [3], is as follows:

$$\begin{aligned} &\min_{\theta, \lambda} z = \theta \\ &s.t.\ \theta x_o - X\lambda \geq 0 \\ &\quad Y\lambda \geq y_o \\ &\quad e\lambda \geq 1 \\ &\quad \lambda \geq 0 \end{aligned} \qquad (IRS-I)$$

3.9. Ouput-oriented increasing returns-to-scale model

The formulation of the output-oriented increasing returns-to-scale (IRS) model, is as follows:

$$\begin{aligned} &\max_{\eta, \lambda} z = \eta \\ &s.t.\quad X\lambda \leq x_o \\ &\quad \eta y_o - Y\lambda \leq 0 \\ &\quad e\lambda \geq 1 \\ &\quad \lambda \geq 0 \end{aligned} \qquad (IRS-O)$$

3.10. Input-oriented decreasing returns-to-scale model

The formulation of the input-oriented decreasing returns-to-scale (DRS) model, described in [3], is as follows:

$$
\begin{aligned}
&\min_{\theta,\lambda} z = \theta \\
&s.t.\ \theta x_o - X\lambda \geq 0 \\
&\qquad Y\lambda \geq y_o \\
&\qquad 0 \leq e\lambda \leq 1 \\
&\qquad \lambda \geq 0
\end{aligned}
\qquad (DRS-I)
$$

3.11. Ouput-oriented decreasing returns-to-scale model

The formulation of the output-oriented decreasing returns-to-scale (DRS) model, is as follows:

$$
\begin{aligned}
&\max_{\eta,\lambda} z = \eta \\
&s.t. \qquad X\lambda \leq x_o \\
&\qquad \eta y_o - Y\lambda \leq 0 \\
&\qquad 0 \leq e\lambda \leq 1 \\
&\qquad \lambda \geq 0
\end{aligned}
\qquad (DRS-O)
$$

3.12. Input-oriented non-radial super efficiency model

The aim of the "super efficiency" DEA models is to provide discriminating characteristics among DMUs, in situations where a significant number of DMUs are efficient. Moreover, super efficiency DEA models using slacks based measure have several benefits, such as unit invariant efficiency scores. The input-oriented non-radial super efficiency model, under the constant returns-to-scale assumption presented in [20], is as follows:

$$
\begin{aligned}
&\min_{\phi,\lambda} z = 1 + \frac{1}{m}\sum_{i=1}^{m} \phi_i \\
&s.t.\ \sum_{j=1,\neq o}^{n} x_{ij}\lambda_j - x_{io}\phi_i \leq x_{io}\ (i = 1,...,m) \\
&\qquad \sum_{j=1,\neq o}^{n} y_{rj}\lambda_j \geq y_{ro}\ (r = 1,...,s) \\
&\qquad \varphi_i \geq 0(\forall i), \lambda_j \geq 0(\forall j)
\end{aligned}
\qquad (SuperSBM-I-C)
$$

3.13. Output-oriented non-radial super efficiency model

The formulation of the output-oriented non-radial super efficiency model, under the constant returns-to-scale assumption, is as follows:

$$
\begin{aligned}
&\min_{\psi,\lambda} z = \frac{1}{1-\frac{1}{s}\sum_{r=1}^{s}\psi_r} \\
&s.t.\ \sum_{j=1,\neq o}^{n} x_{ij}\lambda_j \leq x_{io}\ (i = 1,...,m) \\
&\qquad \sum_{j=1,\neq o}^{n} y_{rj}\lambda_j + y_{ro}\psi_r \geq y_{ro}\ (r = 1,...,s) \\
&\qquad \psi_r \geq 0(\forall r), \lambda_j \geq 0(\forall j)
\end{aligned}
\qquad (SuperSBM-O-C)
$$

3.14. Input-oriented radial super efficiency model

The two previous sub-sections can be extended for variable returns-to-scale models. The first super efficiency DEA model with radial characteristics was introduced in [1]. The formulation of the input-oriented radial super efficiency model, under the variable returns-to-scale assumption, is as follows:

$$\begin{aligned}
&\min z = \theta \\
&s.t.\ \theta x_{io} \geq \sum_{j=1,\neq o}^{n} \lambda_j x_{ij}\ (i = 1, ..., m) \qquad (SuperRadial - I - V) \\
&\quad y_{ro} \leq \sum_{j=1,\neq o}^{n} \lambda_j y_{rj}\ (r = 1, ..., s) \\
&\quad \sum_{j=1,\neq o}^{n} \lambda_j = 1 \\
&\quad \lambda_j \geq 0 (\forall j)
\end{aligned}$$

3.15. Ouput-oriented radial super efficiency model

The formulation of the ouput-oriented radial super efficiency model, under the variable returns-to-scale assumption, is as follows:

$$\begin{aligned}
&\min z = \frac{1}{\eta} \\
&s.t.\ x_{io} \geq \sum_{j=1,\neq o}^{n} \lambda_j x_{ij}\ (i = 1, ..., m) \qquad (SuperRadial - O - V) \\
&\quad \eta y_{ro} \leq \sum_{j=1,\neq o}^{n} \lambda_j y_{rj}\ (r = 1, ..., s) \\
&\quad \sum_{j=1,\neq o}^{n} \lambda_j = 1 \\
&\quad \lambda_j \geq 0 (\forall j)
\end{aligned}$$

4. AMPL IMPLEMENTATION DETAILS

The proposed DEA software package features a variety of DEA models, implemented in AMPL modeling language as separate files with extension ".mod". All the required data are separated from the corresponding model files and need to be stored in a spreadsheet file. Since a ranking of the DEA efficiency scores of k DMUs requires the solution of k Linear Problems (LPs), it is much more convenient to use AMPL script files. These sophisticated script files apart from utilizing the "model" and "data" commands in order to read the files having the model declarations and the data statements respectively, they can also run repetitive AMPL commands. Furthermore, these script files have extension ".run" and usually contain a variety of options, (e.g., selection of appropriate solver, precision, rounding) and/or contain the required commands for producing formatted solution reports (e.g., slack variables, θ, λ values).

Initially, the user needs to fill the data into a pre-formatted spreadsheet file, as it is depicted in figure 1. This spreadsheet file has a few pre-defined named ranges, using the "name manager" facility of MS Excel. The scope of these named ranges is to store the data of the AMPL set regarding the n DMUs, the matrices containing the input X and output Y variables, as also their cardinality numbers m and s respectively. These named ranges allow the interoperability of AMPL with the relational database tables in MS Excel, using the Microsoft Open Database Connectivity (ODBC) driver. More detailed instructions about the communication of the handler supports packages via the ODBC driver can be found in the book by Fourer et al. [10].

An example of the source code of the input-oriented Slacks-based measure model file, presented in the subsection 3.5 is given below.

```
# Model file for the input-oriented Slacks-based measure model

#-----------------------------------------------
# Set Definition

set DMU ordered;

#-----------------------------------------------
# Parameter Definition

param m >= 0;
param s >= 0;
param X{DMU,1..m} >= 0;
```

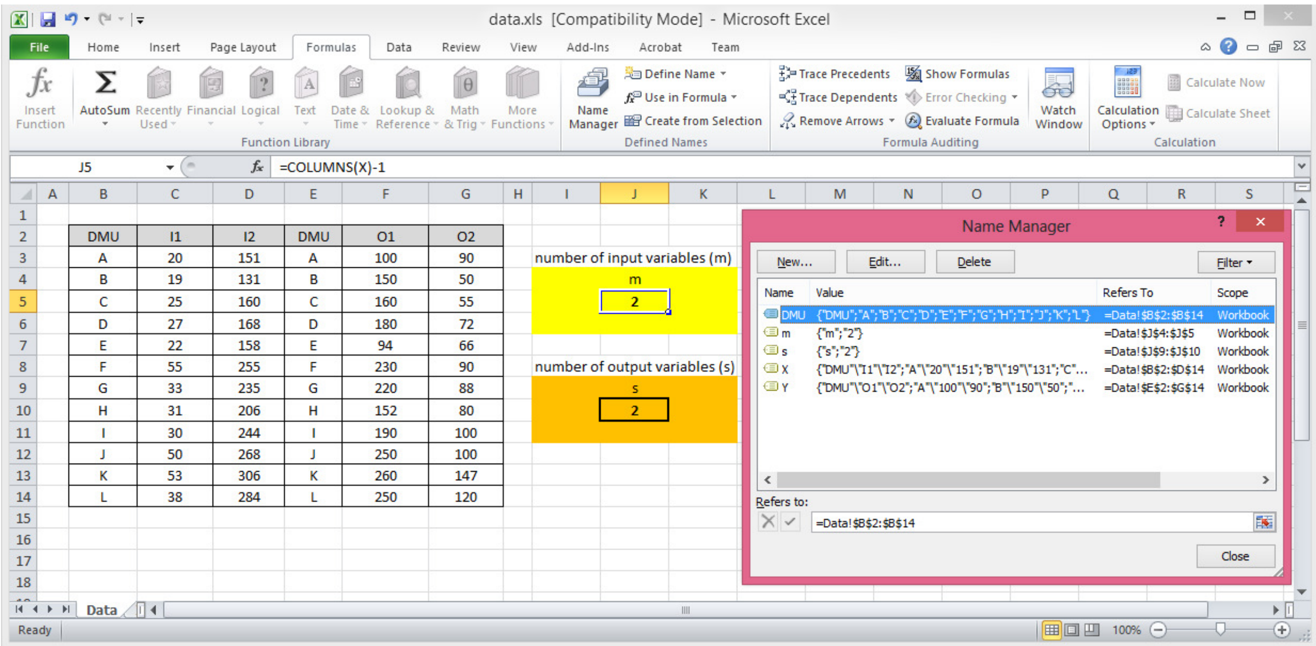

Figure 1 Example of a spreadsheet file using named ranges

```
param Y{DMU,1..s} >= 0;
param iDMU symbolic default first(DMU);

#--------------------------------------------------
# Variable Definition

var lamda{DMU} >= 0;
var s_minus{1..m} >= 0;

#--------------------------------------------------
# Objective Function

minimize efficiency : 1 - (1/m)*sum{i in 1..m}s_minus[i] / X[iDMU,i];

#--------------------------------------------------
# Constraints

subject to input_constraint{j in 1..m}:
X[iDMU,j] = sum {n in DMU} X[n,j] * lamda[n] + s_minus[j];

subject to DMU_constraint{j in 1..s}:
Y[iDMU,j] <= sum {n in DMU} Y[n,j] * lamda[n];
```

An example of the source code of the script file that i) reads the above model file and the data stored in a spreadsheet file named "data.xls", ii) selects the Gurobi solver, and iii) displays the efficiency scores (objective values) and the decision variables (i.e., slack variables and λ values) is given below. The execution of this script file will solve sequentially k LPs (12 in our example).

```
# Script file for the input-oriented Slacks-based measure model

model SBM-I-C.mod;

table m IN "ODBC" "data.xls" : [], m;
read table m;

table s IN "ODBC" "data.xls" : [], s;
read table s;

table DMU IN "ODBC" "data.xls" "DMU" : DMU <- [DMU];
read table DMU;

table X IN "ODBC" "data.xls" :
```

```
[p ~ DMU], {h in 1..m} < X[p,h] ~('I' & h)>;
read table X;

table Y IN "ODBC" "data.xls" :
[p ~ DMU], {h in 1..s} < Y[p,h] ~('O' & h)>;
read table Y;

option solver gurobi_ampl;

for {p in DMU} {
solve;
option omit_zero_rows 1;
display lamda;
display s_minus;

if iDMU==last(DMU)then break;
 let iDMU:=next(iDMU,DMU);
}
```

The solution report contains information regarding the efficiency score, the reference set (λ), and the projection for each DMU. A part of this report concerning the third (C) DMU of our example, is shown below.

```
Gurobi 5.5.0: optimal solution; objective 0.8521654412
4 simplex iterations
lamda [*] :=
A  0.0294118
B  1.04706
;

s_minus [*] :=
1   4.51765
2  18.3941
;
```

As it can be seen from the above results, the efficiency score of the C DMU was 0.8521654412. The reference set gives us the opportunity to identify the efficient DMUs and the extent to which they operate similarly to a given inefficient DMU. The peers of the C DMU are those efficient DMUs that define the facet of the frontier against which the (inefficient) C DMU is projected. Thus, the reference set of this DMU comprises DMUs A and B. Furthermore, the input excesses s^{-} are also displayed in the above results.

In order to verify the correctness of the results, real world datasets have been examined and solved using the well-known commercial DEA Solver Pro v8.0c by SAITECH, Inc. In all these cases, the same results have been obtained, except minor rounding differences.

5. CONCLUSION AND FUTURE WORK

It is well-known that algebraic modeling language are ideal tools for rapid prototyping and optimization model development [11]. Thus, our proposed DEA software package utilizes the flexibility and convenience of the AMPL modeling language. A strong point of the proposed work is the variety of the implemented DEA models either with constant returns-to-scale, variable returns-to-scale characteristics, increasing or decreasing returns-to-scale characteristics, and also their input or output oriented versions.

Since this is a work in progress, new models are still being added to the proposed library. A future research direction is to improve some implementation issues regarding, e.g., any imbalance in data magnitudes, treat problems with nonpositive values (for ratio models) and others. Such guidelines for the preparation of data for DEA problems can be found in [16]. Furthermore, the proposed DEA software library will be soon publicly available by the authors.

References

1 Per Andersen & Niels Christian Petersen (1993). A procedure for ranking efficient units in data envelopment analysis. Management Science, 39(10), 1261–1264.

2 Banker, R. D. , A. Charnes & W. W. Cooper (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science, 30(9), 1078–1092.

3 Rajiv D. Banker & R.M. Thrall (1992). Estimation of returns to scale using data envelopment analysis. European Journal of Operational Research, 62(1), 74–84.

4 Richard S. Barr (2004). DEA software tools and technology. In William W. Cooper, Lawrence M. Seiford & Joe Zhu, editors, *Handbook on Data Envelopment Analysis*, volume 71 of *International Series in Operations Research & Management Science*, pages 539–566. Springer US.

5 Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European Journal of Operational Research, 2(6), 429–444.

6 Wade D. Cook & Larry M. Seiford (2009). Data envelopment analysis (DEA) – thirty years on. European Journal of Operational Research, 192(1), 1–17.

7 William Wager Cooper, Lawrence M. Seiford & Kaoru Tone (2007). Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software. Springer Science+Business Media, LLC, New York, NY, 2nd edition.

8 Emrouznejad, A. (2000). An extension to SAS/OR for decision system support. In Proceedings of the 25th SAS Users Group International Conference. SAS Ins: 259.

9 Michael C. Ferris & Meta M. Voelker (2002). Slice models in general purpose modeling systems: An application to DEA. Optimization Methods and Software, 17(6), 1009–1032.

10 R. Fourer, D.M. Gay & B.W. Kernighan (2002). AMPL: A Modeling Language for Mathematical Programming. Duxbury Press, 2nd edition.

11 Green, R. H. (1996). DIY DEA: Implementing data envelopment analysis in the mathematical programming language AMPL. Omega, 24(4), 489–494.

12 Kalvelagen, E. (2004). Efficiency solving DEA models with GAMS. GAMS Development Corporation, Washington DC, USA.

13 Andreas Kleine & Dennis Sebastian (2005). Generalized DEA-range adjusted measurement. volume 2004 of Operations Research Proceedings, pages 385–392. Springer Berlin Heidelberg.

14 O.B. Olesen & N.C. Petersen (1996). A presentation of GAMS for DEA. Computers & Operations Research, 23(4), 323–339.

15 SAITECH Inc. DEA Solver newsletter no. 9, 2013. Last accessed on May, 14, 2013. URL: http://www.saitech-inc.com/security/NewsLetter9.zip.

16 Joe Sarkis (2007). Preparing your data for DEA. In Joe Zhu & Wade D. Cook, editors, Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis, pages 305–320. Springer US.

17 Stiakakis, E. & Sifaleras, A. (2010). Evaluating the performance of decision making units in the food production industry. In B. Manos, K. Paparrizos, N. Matsatsinis, and J. Papathanasiou, editors, Decision Support Systems in Agriculture, Food and the Environment: Trends, Applications and Advances, pages 173–192. IGI Global, Hershey, PA, USA.

18 Stiakakis, E. & Sifaleras, A. (2013). Combining the priority rankings of DEA and AHP methodologies: A case study on an ICT industry. International Journal of Data Analysis Techniques and Strategies, 5(1), 101–114.

19 Kaoru Tone (2001). A slacks-based measure of efficiency in data envelopment analysis. European Journal of Operational Research, 130(3), 498–509.

20 Kaoru Tone (2002). A slacks-based measure of super-efficiency in data envelopment analysis. European Journal of Operational Research, 143(1), 32–41.

21 Henry Tulkens (1993). On FDH efficiency analysis: Some methodological issues and applications to retail banking, courts, and urban transit. Journal of Productivity Analysis, 4(1-2), 183–210.

22 Paul W. Wilson (2008). FEAR: A software package for frontier efficiency analysis with R. Socio-Economic Planning Sciences, 42(4), 247–254.

23 Joe Zhu (2009). Quantitative models for performance evaluation and benchmarking: data envelopment analysis with spreadsheets and DEA excel solver, volume 51. Kluwer academic publishers, 2nd edition.