MODELING PERFORMANCE MEASUREMENT
Applications and Implementation Issues in DEA

MODELING PERFORMANCE MEASUREMENT

Applications and Implementation Issues in DEA

by

Wade D. Cook
York University, Canada

Joe Zhu
Worcester Polytechnic Institute, U.S.A.

Springer

Wade D. Cook
York University
Canada

Joe Zhu
Worcester Polytechnic Institute
USA

Library of Congress Cataloging-in-Publication Data
A C.I.P. Catalogue record for this book is available
from the Library of Congress.

MODELING PERFORMANCE MEASUREMENT: Applications and Implementation Issues in DEA

ISBN 0-387-24137-X e-ISBN 0-387-24138-8

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11052760

springeronline.com

To the Memory
of
Marsha Jeanne Cook

CONTENTS

Chapter 10 Resource Constrained DEA

Chapter 11 Multicomponent Efficiency: Measurement and Core Business Identification in Multiplant Firms

Chapter 12 Implementation of Robotics: Identifying Efficient Implementors

Chapter 16 Context-Dependent DEA: Models and Extension

Chapter 17 Evaluating Power Plant Efficiency: Hierarchical Models

PREFACE

Data Envelopment Analysis (DEA) is a data-oriented approach for performance evaluation and improvement. In recent years, we have observed a notable increase in interest in DEA techniques and their applications. Basic DEA models and techniques have been well documented in the DEA literature. Although these basic DEA models are useful in determining the best-practice frontier, identification of best-practices is seldom the ultimate goal with respect to performance evaluation. It is generally important to further analyze the business operations after the identification of best-practice, so that in-depth managerial information can be derived. It is also important to correctly design and model the performance issues. Because of the complexity of the business or engineering operations which are often characterized by multiple functions, multiple stages and multiple levels, new (and advanced) DEA methods are needed to reconcile the multidimensional aspects of performance evaluation issues.

The book presents unified results from the authors' recent DEA research. New methodologies and techniques are developed in application-driven scenarios, to go beyond identification of the best-practice frontier, and seek solutions to aid managerial decisions. These new DEA developments are deeply grounded in real-world applications. DEA researchers and practitioners alike will find this book helpful. Theory is provided for DEA researchers for further development and possible extensions. However, each theory is also presented in a practical way for DEA practitioners via numerical examples, simple real management cases and verbal descriptions.

The book covers pure DEA applications in such areas as highway maintenance, technology implementations, and others. DEA methodology enhancements are wrapped into applications. New DEA theoretical developments are included, for example, on how to use DEA as a benchmarking tool, and how to use DEA in multi-criteria decision making. The book provides a balanced coverage of DEA for both academic researchers and industry practitioners. It addresses advanced/new DEA methodology and techniques that are developed for modeling unique and new performance evaluation issues. Some of the DEA models can be computed using the accompanying DEAFrontier software which is an Excel Add-In.

Wade D. Cook
Schulich School of Business
York University
4700 Keele Street
Toronto, Ontario
Canada M3J 1P3
Email: wcook@schulich.yorku.ca

Joe Zhu
Department of Management
Worcester Polytechnic Institute
Worcester, MA 01609 USA
Email: jzhu@wpi.edu

Chapter 1

DATA ENVELOPMENT ANALYSIS

1.1. INTRODUCTION

Data Envelopment Analysis (DEA) is a relatively new "data oriented" approach for evaluating the performance of a set of peer entities called Decision Making Units (DMUs) which convert multiple inputs to multiple outputs. The definition of a DMU is generic and flexible. Recent years have seen a great variety of applications of DEA for use in evaluating the performances of many different kinds of entities engaged in many different activities in many different contexts in many different countries. These DEA applications have used DMUs of various forms to evaluate the performance of entities, such as hospitals, US Air Force wings, universities, cities, courts, business firms, and others, including the performance of countries, regions, etc. Because it requires very few assumptions, DEA has also opened up possibilities for use in cases which have been resistant to other approaches because of the complex (often unknown) nature of the relations between the multiple inputs and multiple outputs involved in DMUs (Cooper, Seiford and Zhu, 2004).

Since DEA in its present form was first introduced in 1978, researchers in a number of fields have quickly recognized that it is an excellent and easily used methodology for modeling operational processes for performance evaluations (Cooper, Seiford and Tone, 2000). This has been accompanied by other developments. For instance, Zhu (2002) provides a number of DEA spreadsheet models that can be used in performance evaluation and benchmarking. DEA's empirical orientation and the absence of a need for the numerous *a priori* assumptions that accompany other approaches (such as standard forms of statistical regression analysis) have resulted in its use in a number of studies involving efficient frontier estimation in the governmental and nonprofit sector, in the regulated sector, and in the private sector.

In their originating study, Charnes, Cooper, and Rhodes (1978) described DEA as a 'mathematical programming model applied to observational data [that] provides a new way of obtaining empirical estimates of relations - such as the production functions and/or efficient production possibility surfaces – that are cornerstones of modern economics'.

Formally, DEA is a methodology directed to frontiers rather than central tendencies. Instead of trying to fit a regression plane through the *center* of the data as in statistical regression, for example, one 'floats' a piecewise linear surface to rest on top of the observations. Because of this perspective, DEA proves particularly adept at uncovering relationships that remain hidden from other methodologies. For instance, consider what one wants to mean by "efficiency", or more generally, what one wants to mean by saying that one DMU is more efficient than another DMU. This is accomplished in a straightforward manner by DEA without requiring expectations and variations with various types of models such as in linear and nonlinear regression models.

1.2. ENVELOPMENT AND MULTIPLIER DEA MODELS

Consider a set of n observations on the DMUs. Each observation, DMU_j ($j = 1, \ldots, n$), uses m inputs x_{ij} ($i = 1, 2, \ldots, m$) to produce s outputs y_{rj} ($r = 1, 2, \ldots, s$). The CCR ratio model can be expressed as

$$\max h_o(u,v) = \sum_r u_r y_{ro} / \sum_i v_i x_{io} \tag{1.1}$$

where the variables are the u_r's and the v_i's and the y_{ro}'s and x_{io}'s are the observed output and input values, respectively, of $\boldsymbol{DMU_o}$, the DMU to be evaluated. Of course, without further additional constraints (developed below) (1.1) is unbounded.

A set of normalizing constraints (one for each DMU) reflects the condition that the virtual output to virtual input ratio of every DMU, including $DMU_j = DMU_o$, must be less than or equal to unity. The mathematical programming problem may thus be stated as

$$\begin{aligned} & \max h_o(u,v) = \sum_r u_r y_{ro} / \sum_i v_i x_{io} \\ & \text{subject to:} \\ & \sum_r u_r y_{rj} / \sum_i v_i x_{ij} \le 1 \text{ for } j = 1, \ldots, n, \\ & u_r, v_i \ge 0. \end{aligned} \tag{1.2}$$

The above ratio form yields an infinite number of solutions; if (u^*, v^*) is optimal, then (αu^*, αv^*) is also optimal for $\alpha > 0$. However, the transformation developed by Charnes and Cooper (1962) for linear fractional programming selects a representative solution [i.e., the solution (u, v) for which $\sum_{i=1}^{m} v_i x_{io} = 1$] and yields the equivalent linear programming problem [the change of variables from (u, v) to (μ, v) is a result of the Charnes-Cooper transformation],

$$
\begin{aligned}
& \max z = \sum_{r=1}^{s} \mu_r y_{ro} \\
& \text{subject to} \\
& \sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \le 0 \\
& \sum_{i=1}^{m} v_i x_{io} = 1 \\
& \mu_r, v_i \ge 0
\end{aligned}
\tag{1.3}
$$

The dual program of (1.3) is

$$
\begin{aligned}
& \theta^* = \min \theta \\
& \text{subject to} \\
& \sum_{j=1}^{n} x_{ij} \lambda_j \le \theta x_{io} \qquad i = 1,2,\ldots,m; \\
& \sum_{j=1}^{n} y_{rj} \lambda_j \ge y_{ro} \qquad r = 1,2,\ldots,s; \\
& \lambda_j \ge 0 \qquad j = 1,2,\ldots,n.
\end{aligned}
\tag{1.4}
$$

Since $\theta = 1$ is a feasible solution to (1.4), the optimal value to (1.4), $\theta^* \le 1$. If $\theta^* = 1$, then the current input levels cannot be reduced (proportionally), indicating that DMU_o is on the frontier. Otherwise, if $\theta^* < 1$, then DMU_o is dominated by the frontier. θ^* represents the (input-oriented) efficiency score of DMU_o.

Table 1-1. Supply Chain Operations Within a Week

DMU	Cost ($100)	Response time (days)	Profit ($1,000)
1	1	5	2
2	2	2	2
3	4	1	2
4	6	1	2
5	4	4	2

We now consider a simple numerical example shown in Table 1.1 where we have five DMUs (supply chain operations). Within a week, each DMU generates the same profit of $2,000 with a different combination of supply chain cost and response time.

Figure 1-1 presents the five DMUs and the piecewise linear frontier. DMUs 1, 2, 3, and 4 are on the frontier. If we calculate model (1.4) for DMU5,

$$\begin{aligned}
&\text{Min } \theta \\
&\text{Subject to:} \\
&1\,\lambda_1 + 2\lambda_2 + 4\lambda_3 + 6\lambda_4 + 4\lambda_5 \le 4\theta \\
&5\,\lambda_1 + 2\lambda_2 + 1\lambda_3 + 1\lambda_4 + 4\lambda_5 \le 4\theta \\
&2\,\lambda_1 + 2\lambda_2 + 2\lambda_3 + 2\lambda_4 + 2\lambda_5 \ge 2 \\
&\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \ge 0
\end{aligned}$$

we obtain a set of unique optimal solutions of $\theta^* = 0.5$, $\lambda_2^* = 1$, and $\lambda_j^* = 0$ ($j \neq 2$), indicating that DMU2 is the benchmark for DMU5, and DMU5 should reduce its cost and response time to the amounts used by DMU2.

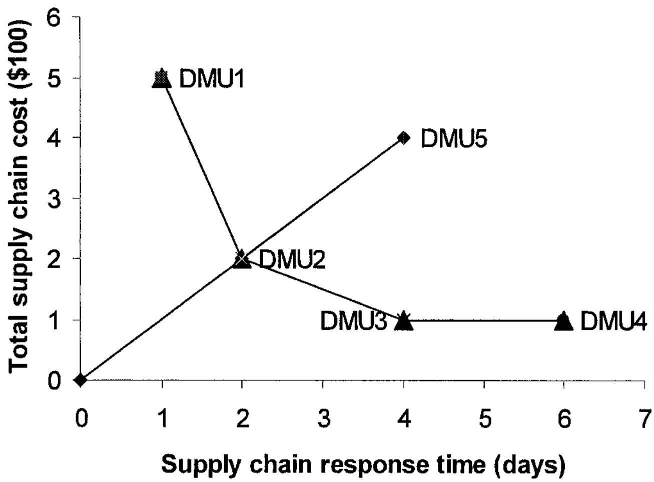

Figure 1-1. Five Supply Chain Operations

Now, if we calculate model (1.4) for DMU4, we obtain $\theta^* = 1$, $\lambda_4^* = 1$, and $\lambda_j^* = 0$ ($j \neq 4$), indicating that DMU4 is on the frontier. However, Figure 1-1 indicates that DMU4 can still reduce its response time by 2 days to reach DMU3. This individual input reduction is called input slack.

In fact, both input and output slack values may exist in model (1.4). Usually, after calculating (1.4), we have

$$\begin{cases} s_i^- = \theta^* x_{io} - \sum_{j=1}^{n} \lambda_j x_{ij} & i = 1,2,...,m \\ s_r^+ = \sum_{j=1}^{n} \lambda_j y_{rj} - y_{ro} & r = 1,2,...,s \end{cases} \tag{1.5}$$

where s_i^- and s_r^+ represent input and output slacks, respectively. An alternate optimal solution of θ^* = 1 and λ_3^* = 1 exists when we calculate model (1.4) for DMU4. This leads to s_1^- = 2 for DMU4. However, if we obtain θ^* = 1 and λ_4^* = 1 from model (1.4), we have all zero slack values. i.e., because of possible multiple optimal solutions, (1.4) may not yield all the non-zero slacks.

Therefore, we use the following linear programming model to determine the possible non-zero slacks after (1.2) is solved.

$$\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+$$

subject to

$$\begin{aligned} &\sum_{j=1}^{n} x_{ij} \lambda_j + s_i^- = \theta^* x_{io} && i = 1,2,...,m; \\ &\sum_{j=1}^{n} y_{rj} \lambda_j - s_r^+ = y_{ro} && r = 1,2,...,s; \\ &\lambda_j \geq 0 && j = 1,2,...,n. \end{aligned} \tag{1.6}$$

For example, applying (1.6) to DMU4 yields

$$\begin{aligned} &\text{Max } s_1^- + s_2^- + s_1^+ \\ &\text{Subject to} \\ &1\,\lambda_1 + 2\lambda_2 + 4\lambda_3 + 6\lambda_4 + 4\lambda_5 + s_1^- = 6\,\theta^* = 6 \\ &5\,\lambda_1 + 2\lambda_2 + 1\lambda_3 + 1\lambda_4 + 4\lambda_5 + s_2^- = 1\,\theta^* = 1 \\ &2\,\lambda_1 + 2\lambda_2 + 2\lambda_3 + 2\lambda_4 + 2\lambda_5 - s_1^+ = 2 \\ &\lambda_1,\ \lambda_2,\ \lambda_3, \lambda_4,\ \lambda_5,\ s_1^-,\ s_2^-,\ s_1^+ \geq 0 \end{aligned}$$

with optimal slacks of $s_1^{-*} = 2$, $s_2^{-*} = s_1^{+*} = 0$.

Definition 1.1 (DEA Efficiency): The performance of DMU_o is fully (100%) efficient if and only if both (i) θ^* = 1 and (ii) all slacks $s_i^{-*} = s_r^{+*} = 0$.

Definition 1.2 (Weakly DEA Efficient): The performance of DMU_o is weakly efficient if and only if both (i) θ^* = 1 and (ii) s_i^{-*} $\neq$ 0 and/or s_r^{+*} $\neq 0$ for some *i* and *r*.

In Figure 1.2, DMUs 1, 2, and 3 are efficient, and DMU 4 is weakly efficient. (The slacks obtained by (1.6) are called DEA slacks.)

In fact, models (1.4) and (1.6) represent a two-stage DEA process involved in the following DEA model.

$$
\begin{aligned}
&\min \theta - \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j \geq 0 \qquad j = 1,2,\ldots,n.
\end{aligned} \tag{1.7}
$$

The presence of the non-Archimedean ε in the objective function of (1.7) effectively allows the minimization over θ to preempt the optimization involving the slacks, s_i^- and s_r^+. Thus, (1.7) is calculated in a two-stage process with maximal reduction of inputs being achieved first, via the optimal θ^* in (1.4); then, in the second stage, movement onto the efficient frontier is achieved via optimizing the slack variables in (1.6).

In fact, the presence of weakly efficient DMUs is the cause of multiple optimal solutions. Thus, if weakly efficient DMUs are not present, the second stage calculation (1.6) is not necessary, and we can obtain the slacks using (1.5). However, priori to calculation, we usually do not know whether weakly efficient DMUs are present.

Model (1.7) is usually called "envelopment" DEA model. The dual program to (1.7) is called "multiplier" DEA model.

$$
\begin{aligned}
&\max z = \sum_{r=1}^{s} \mu_r y_{ro} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \\
&\sum_{i=1}^{m} v_i x_{io} = 1 \\
&\mu_r, v_i \geq \varepsilon > 0
\end{aligned} \tag{1.8}
$$

If we consider the following DEA model,

$$\begin{aligned}
&\text{Min } \textstyle\sum_i v_i x_{io} / \sum_r u_r y_{ro} \\
&\text{Subject to} \\
&\textstyle\sum_i v_i x_{ij} / \sum_r u_r y_{rj} \geq 1 \text{ for } j = 1, \ldots, n, \\
&u_r,\ v_i \geq \varepsilon > 0.
\end{aligned} \tag{1.9}$$

where $\varepsilon > 0$ is the previously defined non-Archimedean element, then we have the following output-oriented multiplier and envelopment DEA models

$$\begin{aligned}
&\min q = \sum_{i=1}^{m} v_i x_{io} \\
&\text{subject to} \\
&\sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} \geq 0 \\
&\sum_{r=1}^{s} \mu_r y_{ro} = 1 \\
&\mu_r, v_i \geq \varepsilon, \quad \forall r, i
\end{aligned} \tag{1.10}$$

$$\begin{aligned}
&\max \phi + \varepsilon\left(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\right) \\
&\text{subject to:} \\
&\sum_{j=1}^{n} x_{ij}\lambda_j + s_i^- = x_{io} \quad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} y_{rj}\lambda_j - s_r^+ = \phi y_{ro} \quad r = 1,2,\ldots,s; \\
&\lambda_j \geq 0 \qquad\qquad j = 1,2,\ldots,n.
\end{aligned} \tag{1.11}$$

As before, model (1.11) is calculated in a two-stage process. First, we calculate ϕ^* by ignoring the slacks. Then we optimize the slacks by fixing ϕ^* in the following linear programming problem,

$$
\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} x_{ij}\lambda_j + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} y_{rj}\lambda_j - s_r^+ = \phi^* y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j \geq 0 \qquad j = 1,2,\ldots,n.
\end{aligned}
\tag{1.12}
$$

We then modify the previous input-oriented definition of DEA efficiency to the following output-oriented version.

Definition 1.3: DMU_o is efficient if and only if $\phi^* = 1$ and $s_i^{-*} = s_r^{+*} = 0$ for all i and r. DMU_o is weakly efficient if $\phi^* = 1$ and $s_i^{-*} \neq 0$ and (or) $s_r^{+*} \neq 0$ for some i and r.

The frontier determined by the above DEA models exhibits constant returns to scale (CRS). Thus, the above DEA models are called CRS DEA models with different orientations. Figure 1-2 shows a CRS frontier – ray OB. Based upon this CRS frontier, only B is efficient.

The constraint on $\sum_{j=1}^{n}\lambda_j$ in the envelopment models actually determines the returns to scale (RTS) type of an efficient frontier. If we add $\sum_{j=1}^{n}\lambda_j = 1$, we obtain VRS (variable RTS) models. The frontier is ABCD as shown in Figure 1-2.

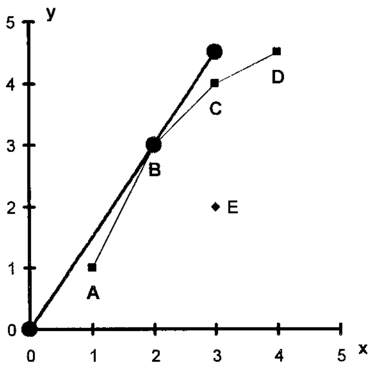

Figure 1-2. CRS Frontier

If we replace $\sum_{j=1}^{n} \lambda_j = 1$ with $\sum_{j=1}^{n} \lambda_j \leq 1$, then we obtain non-increasing RTS (NIRS) envelopment models. In Figure 1-3, the NIRS frontier consists of DMUs B, C, D and the origin.

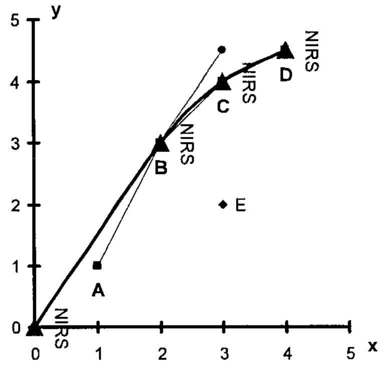

Figure 1-3. NIRS Frontier

If we replace $\sum_{j=1}^{n} \lambda_j = 1$ with $\sum_{j=1}^{n} \lambda_j \geq 1$, then we obtain non-decreasing RTS (NDRS) envelopment models. In Figure 1-4, the NDRS frontier consists of DMUs, A, B, and the section starting with B on ray OB.

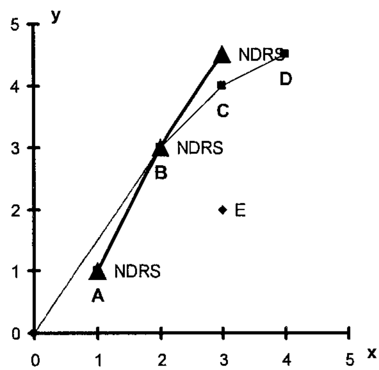

Figure 1-4. NDRS Frontier

Table 1-2 summarizes the envelopment and the multiplier models with respect to the orientations and frontier types. The last row presents the efficient target (DEA projection) of a specific DMU under evaluation.

Table 1-2. DEA Models

Frontier Type	Input-Oriented	Output-Oriented
CRS	$\min \theta - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io}$ $i = 1,2,...,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro}$ $r = 1,2,...,s;$ $\lambda_j \geq 0$ $j = 1,2,...,n.$	$\max \phi + \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+)$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io}$ $i = 1,2,...,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi y_{ro}$ $r = 1,2,...,s;$ $\lambda_j \geq 0$ $j = 1,2,...,n.$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$	
Efficient Target	$\begin{cases} \hat{x}_{io} = \theta^* x_{io} - s_i^{-*} & i = 1,2,...,m \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r = 1,2,...,s \end{cases}$	$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} & i = 1,2,...,m \\ \hat{y}_{ro} = \phi^* y_{ro} + s_r^{+*} & r = 1,2,...,s \end{cases}$

	$\max \sum_{r=1}^{s} \mu_r y_{ro} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} + \mu \leq 0$ $\sum_{i=1}^{m} v_i x_{io} = 1$ $\mu_r, v_i \geq 0(\varepsilon)$	$\min \sum_{i=1}^{m} v_i x_{io} + \nu$ subject to $\sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + \nu \geq 0$ $\sum_{r=1}^{s} \mu_r y_{ro} = 1$ $\mu_r, v_i \geq 0(\varepsilon)$
CRS	where $\mu = 0$	where $\nu = 0$
VRS	where μ free	where ν free
NIRS	where $\mu \leq 0$	where $\nu \geq 0$
NDRS	where $\mu \geq 0$	where $\nu \leq 0$

1.3. ASSURANCE REGION DEA MODELS

Note that the only restriction on the multiplier DEA models is the positivity of the multipliers imposed by ε. In the DEA literature, a number of approaches have been proposed to introduce additional restrictions on the values that the multipliers can assume.

Some of the techniques for enforcing these additional restrictions include imposing upper and lower bounds on individual multipliers (Dyson and Thanassoulis, 1988; Roll, Cook, and Golany, 1991); imposing bounds on ratios of multipliers (Thompson et al., 1986); appending multiplier inequalities (Wong and Beasley, 1990); and requiring multipliers to belong to given closed cones (Charnes et al., 1989).

We here present the assurance region (AR) approach of Thompson et al. (1986). To illustrate the AR approach, suppose we wish to incorporate additional inequality constraints of the following form into the multiplier DEA models as given in Table 1-2:

$$\begin{aligned} &\alpha_i \le \frac{v_i}{v_{i_o}} \le \beta_i, \qquad i = 1,\ldots,m \\ &\delta_r \le \frac{\mu_r}{\mu_{r_o}} \le \gamma_r, \qquad r = 1,\ldots,s \end{aligned} \tag{1.13}$$

Here, v_{i_o} and μ_{r_o} represent multipliers which serve as "numeraires" in establishing the upper and lower bounds represented here by α_i, β_i, and by δ_r, γ_r for the multipliers associated with inputs i =1, ..., m and outputs r = 1, ..., s where $\alpha_{i_o} = \beta_{i_o} = \delta_{r_o} = \gamma_{r_o} = 1$. The above constraints are called Assurance Region (AR) constraints as developed by Thompson et al. (1986) and defined more precisely in Thompson et al. (1990).

Uses of such bounds are not restricted to prices. For example, Zhu (1996a) uses an assurance region approach to establish bounds on the weights obtained from uses of Analytic Hierarchy Processes in Chinese textile manufacturing in order to reflect how the local government in measuring the textile manufacturing performance.

The generality of these AR constraints provides flexibility in use. Prices, utils and other measures may be accommodated and so can mixtures of such concepts. Moreover, one can first examine provisional solutions and then tighten or loosen the bounds until one or more solutions is attained that appears to be reasonably satisfactory to decision makers who cannot state the values for their preferences in an a priori manner.

1.4. SLACK BASED DEA MODELS

The input-oriented DEA models consider the possible (proportional) input reductions while maintaining the current levels of outputs. The output-oriented DEA models consider the possible (proportional) output augmentations while keeping the current levels of inputs. Charnes, Cooper, Golany, Seiford and Stutz (1985) develop an additive DEA model which considers possible input decreases as well as output increases simultaneously. The additive model is based upon input and output slacks. For example,

$$\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to:} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned} \tag{1.14}$$

Note that model (1.8) assumes equal marginal worth for the nonzero input and output slacks. Therefore, caution should be excised in selecting the units for different input and output measures. Some *a priori* information may be required to prevent an inappropriate summation of non-commensurable measures. Previous management experience and expert opinion, which prove important in productivity analysis, may be used (see Seiford and Zhu (1998)).

Model (1.8) therefore is modified to a weighted CRS slack-based model as follows (Ali, Lerme and Seiford, 1995; Thrall, 1996).

$$\begin{aligned}
&\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned} \tag{1.15}$$

where w_i^- and w_r^+ are user-specified weights obtained through value judgment. The DMU_o under evaluation will be termed efficient *if and only if* the optimal value to (1.9) is equal to zero. Otherwise, the nonzero optimal s_i^{-*} identifies an excess utilization of the *i*th input, and the non-zero optimal s_r^{+*} identifies a deficit in the *r*th output. Thus, the solution of (1.15) yields the information on possible adjustments to individual outputs and inputs of each DMU. Obviously, model (1.15) is useful for setting targets for inefficient DMUs with *a priori* information on the adjustments of outputs and inputs.

One should note that model (1.15) does not necessarily yield results that are different from those obtained from the model (1.14). In particular, it will not change the classification from efficient to inefficient (or vice versa) for any DMU.

Model (1.15) identifies a CRS frontier, and therefore is called CRS slack-based model. Table 1.5 summarizes the slack-based models in terms of the frontier types.

Table 1-3. Slack-based Models

Frontier type	Slack-based DEA Model
CRS	$\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \quad i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \quad r = 1,2,\ldots,s;$ $\lambda_j, s_i^-, s_r^+ \geq 0$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$

1.5. MEASURE-SPECIFIC DEA MODELS

Although DEA does not need *a priori* information on the underlying functional forms and weights among various input and output measures, it assumes proportional improvements of inputs or outputs. This assumption becomes invalid when a preference structure over the improvement of different inputs (outputs) is present in evaluating (inefficient) DMUs (see Zhu (1996b)). We need models where a particular set of performance measures is given pre-emptive priority to improve.

Let $I \subseteq \{1,2, \ldots, m\}$ and $O \subseteq \{1,2, \ldots, s\}$ represent the sets of specific inputs and outputs of interest, respectively. Based upon the envelopment models, we can obtain a set of measure-specific models where only the inputs associated with I or the outputs associated with O are optimized (see Table 1-4).

The measure-specific models can be used to model uncontrollable inputs and outputs (see Banker and Morey (1986)). The controllable measures are related to set I or set O.

A DMU is efficient under envelopment models if and only if it is efficient under measure-specific models. i.e., both the measure-specific models and the envelopment models yield the same frontier. However, for inefficient DMUs, envelopment and measure-specific models yield different efficient targets.

Consider Figure 1-1. If the response time input is of interest, then the measure-specific model will yield the efficient target of S1 for inefficient S. If the cost input is of interest, S3 will be the target for S. The envelopment model projects S to S2 by reducing the two inputs proportionally.

Table 1-4. Measure-specific Models

Frontier Type	Input-Oriented	Output-Oriented
CRS	$\begin{array}{ll} \min \theta - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+) & \\ \text{subject to} & \\ \sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = \theta x_{io} & i \in I; \\ \sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} & i \notin I; \\ \sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} & r = 1,2,...,s; \\ \lambda_j \geq 0 & j = 1,2,...,n. \end{array}$	$\begin{array}{ll} \max \phi + \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+) & \\ \text{subject to} & \\ \sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} & i = 1,2,...,m; \\ \sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = \phi y_{ro} & r \in O; \\ \sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} & r \notin O; \\ \lambda_j \geq 0 & j = 1,2,...,n. \end{array}$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$	
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$	
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$	
Efficient Target	$\begin{cases} \hat{x}_{io} = \theta^* x_{io} - s_i^{-*} & i \in I \\ \hat{x}_{io} = x_{io} - s_i^{-*} & i \notin I \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r = 1,2,...,s \end{cases}$	$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} & i = 1,2,...,m \\ \hat{y}_{ro} = \phi^* y_{ro} + s_r^{+*} & r \in O \\ \hat{y}_{ro} = y_{ro} + s_r^{+*} & r \notin O \end{cases}$

1.6. SOLVING DEA WITH DEAFRONTIER SOFTWARE

One can solve the DEA models discussed previously using the spreadsheets and Excel Solver as described in Zhu (2002). In this section, we will demonstrate how to solve the DEA models using the *DEAFrontier* software supplied with the book.

1.6.1 DEAFrontier Software

DEAFrontier is an Add-In for Microsoft® Excel and uses the Excel Solver. This software requires Excel 97 or later versions.

To install the software the CD-ROM using Windows, you may follow these steps:

Step 1. Insert the CD-ROM into your computer's CD-ROM drive. (If the auto run doe not execute, following the following steps.)

Step 2. Launch Windows Explore.
Step 3. Click Browse to browse the CD and find the file "*Setup.exe*".
Step 4: Run "Setup.exe"

DEAFrontier does not set any limit on the number of units, inputs or outputs. However, please check www.solver.com for problem sizes that various versions of Excel Solver can handle (see Table 1-5).

Table 1-5. Microsoft® Excel Solver Problem Size

Problem Size:	Standard Excel Solver	Premium Solver	Premium Solver Platform
Variables x Constraints	200 x 200	1000 x 8000	2000 x 8000

Source: www.solver.com

To run *DEAFrontier*, the Excel Solver must first be installed, and the Solver parameter dialog box must be displayed at least once in the Excel session. Otherwise, an error may occur when you run the software, as shown in Figure 1-5. (*Please also make sure that the Excel Solver works properly. One can use the file "solvertest.xls" to test whether the Excel Solver works. This test file is also available at www.deafrontier.com/solvertest.xls.*)

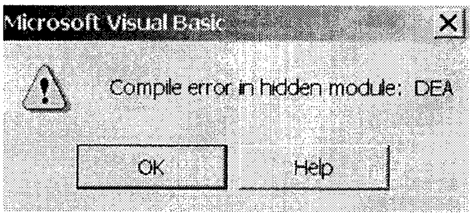

Figure 1-5. Error Message

You may follow the following steps.

First, in Excel, invoke the Solver by using the Tools/Solver menu item as shown in Figure 1-6. This will load the Solver parameter dialog box as shown in Figure 1-7. Then close the Solver parameter dialog box by clicking the Close button. Now, you have successfully loaded the Excel Solver.

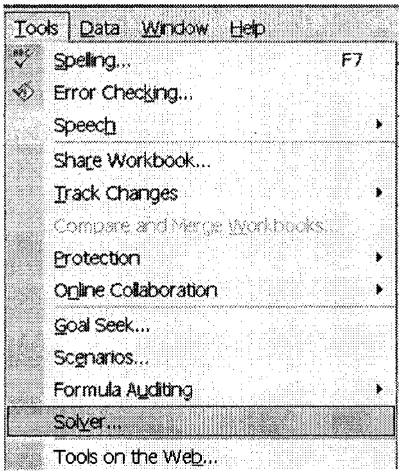

Figure 1-6. Display Solver Parameters Dialog Box

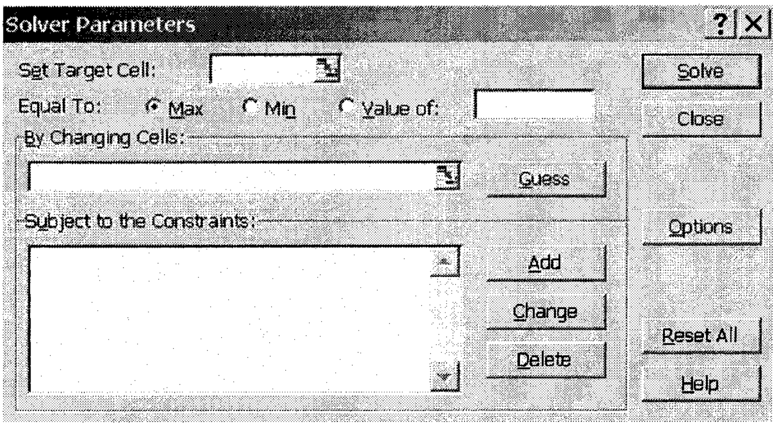

Figure 1-7. Solver Parameters Dialog Box

If Solver does not exist in the Tools menu, you need to select Tools/Add-Ins, and check the Solver box, as shown in Figure 1-8. (If Solver does not show in the Add-Ins, you need to install the Solver first.)

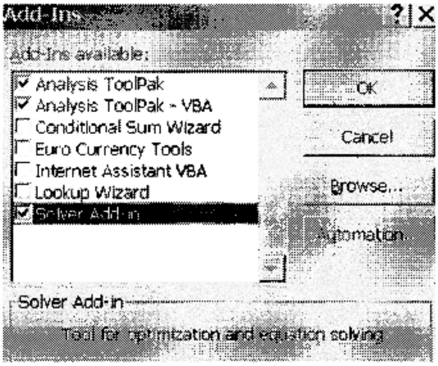

Figure 1-8. Solver Add-In

Next, open the file DEAFrontier.xla, and a "DEAFrontier" menu is added at the end of the Excel menu. (see Figure 1-9). Now, the *DEAFrontier* software is ready to run.

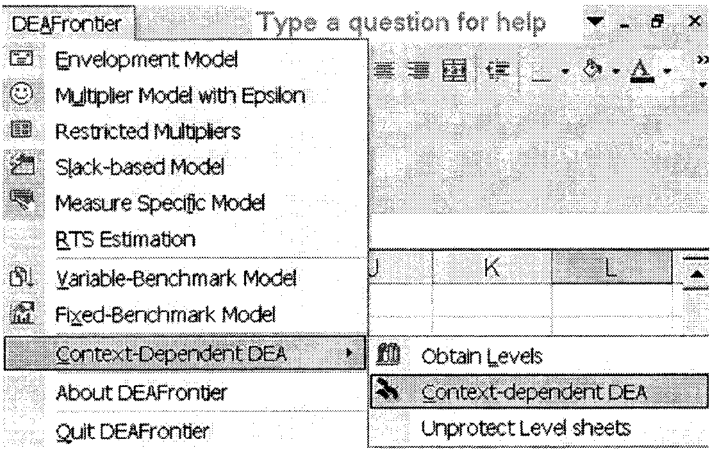

Figure 1-9. DEAFrontier Menu

1.6.2 Organize the Data

The sheet containing the data for DMUs under evaluations must be named as "Data". The data sheet should have the format as shown in Figure 1-10.