Applications of DEA to Software Engineering Management

By

David N. Reese

Centre for Management of Technology and Entrepreneurship
Faculty of Applied Sciences and Engineering
University of Toronto
4 Taddle Creek Road
Toronto, Ontario
M5S 3G9, Canada

A Thesis submitted in conformity with the requirements for the Degree of Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-28044-6

Canada

Abstract

"Applications of DEA to Software Engineering Management"

by David N. Reese

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto
1997

Software systems play an increasingly important role in organizational effectiveness, as well as in gaining competitive advantage and in differentiating organizations from their competition. This applies to both producers of software and those organizations that utilize software. Thus, our ability to efficiently produce high quality software is a crucial factor in determining the success of many organizations. Unfortunately, the track record of most software producers remains poor, with only a handful of notable exceptions worldwide. In addition to new and improved methods and technology, many experts have cited the need to overcome the difficulties associated with managing the software engineering process as solutions to this so-called 'software crisis'.

This thesis addresses several limitations of DEA techniques that can arise through their application to software engineering management. New theoretical contributions are made in three main areas. The first area is crucial to the performance measurement process. New and enhanced models of software production are presented which divide the software production process into multiple phases and are capable of evaluating data sets containing projects with varying degrees of new and modified code.

Measuring overall efficiency and effectiveness is fundamental to the management control process. This is the motivation for the second area: researching the relationship between traditional economic production measures (and definitions) and DEA multiplier flexibility. A prescriptive framework for the application of DEA models to measure overall efficiency and effectiveness is presented along with several new DEA models important to this framework.

The third area is related to the application of DEA to software project planning and presents new tools for forecasting and trade-off analysis. Existing DEA techniques are adapted for the purpose of conducting general trade-off analysis. Inherent in this analysis process is the generation of efficient project forecasts.

This document, while inspired by software engineering management, contains new theoretical contributions which are not limited in their application to this domain only. In particular, this applies to the framework to measure overall efficiency and the new measures that it contains. The new methods of forecasting and trade-off analysis are also applicable to domains where such project management tools are appropriate.

To Amanda

Acknowledgments

I would like to express my gratitude to Professor Joseph C. Paradi for his supervision, teachings and mentorship. He was always graciously made the time to discuss research issues, the 'big picture' or anything else on my mind. I must gratefully acknowledge the teachings, support and motivation of Professor Lawrence Seiford. He bears much of the responsibility for my strong interest in DEA. I would also like to acknowledge the efforts of Professor Wade Cook, Professor Sue Easun, Professor Mark Fox, and Professor I. B. Turksen. I am grateful for their time and valuable comments.

I would like to thank the corporate sponsors of this research, the Toronto-Dominion Bank and Ernst and Young. Both institutions provided information, experience and financial support. I would especially like to thank Larry from Ernst and Young, and Don from the Toronto-Dominion Bank for their valuable insights.

My sincere thanks is also extended to Claire, Gloria, Henry, Don, John and the rest of the members of the CMTE for their support, feedback and wonderful company.

Lastly, but most importantly, I would like to thank Amanda for her love, support and patience; and my parents Ken and Patricia for their motivation and encouragement in all my endeavors.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 *Introduction*

"Those who fall in love with practice without science are like a sailor who steers a ship without a helm or compass, and who never can be certain whither he is going." Leonardo Da Vinci

1.1. Background

Information technology has quickly assumed a very central role in enterprise effectiveness as well as in gaining and maintaining competitive advantage. A crucial component of information technology, the software system, has the potential to deliver tremendous business value. Software is, after all, the component that typically represents the "difference" between a firm and its competition.

Recent estimates of software expenditures indicate that over $500,000,000,000US is spent annually on worldwide software production [WORT94]. Moreover, software is reported as mission critical in a rapidly growing number of industries and plays an increasingly important role in successful organizations [MERL94]. The size of these software systems is also doubling every 5 to 10 years [GIBB94]. Unfortunately, as the importance and influence of software greatly increases, our ability to produce it does not.

Over a decade ago, it was reported that over 25% of large development projects fail to deliver anything. Alarmingly, this situation has not improved since recent literature reports that nearly three quarters of all large systems are deemed to be operating failures, that either do not perform as intended or are scrapped

[GIBB94]. Furthermore, a recent study by Ernst and Young reports that 46% of software systems are never completed, 24% are delivered but never installed, 22% are installed but substantially modified within a year, leaving only 8% in use without modification [MERL94].

Many leading experts have called for an engineering approach to the development of software, combined with the continuous improvement of techniques and tools, as the solution to the problem [PRES92]. Others have argued that organizations have responded with expensive technological solutions as opposed to developing a sound infrastructure, often ignoring the need to manage this change [MERL94]. The problems associated with managing the software producing organization have also been recognized in the literature as an area where serious problems do exist, but where the opportunities are for the greatest improvements (*c.f.* [HUMP89], [SAGE95]).

An important underpinning of any engineering discipline is objective measurement and analysis. Software engineering is no exception. Here, performance measurement is necessary and required to gauge process improvement and to properly evaluate techniques and tools. Furthermore, proper measurement and analysis play a crucial role in engineering management and control. Thus, measurement and analysis serves as an integral part of the suggested solutions to the software problem.

1.2. Related Literature

One can view the management control function as consisting of *planning* and *control* processes that occur at all levels of the organization (*c.f.* [ANTH89]). Planning can be roughly described as "deciding what to do" and control described as "assuring the implementation of plans". Furthermore, one can characterize management's use of measurement and analysis techniques as being applied for planning and control purposes.

Data Envelopment Analysis, or DEA, is well established as a powerful tool for the measurement of *productive efficiency*. These efficiency measures are comparisons of actual producer performance to the best practice performance of its peers. One of the most important aspects of DEA is the ability to measure and analyze the performance of multi-dimensional production processes. This ability makes DEA an ideal management tool for measuring the efficiency of software production: a complex multiple input - multiple output process. DEA methods are very versatile and can be applied as management tools for both planning and control.

In [REES93] and [PARA95], the authors compare DEA to the traditional performance ratio approaches of measuring software project performance. They demonstrate that DEA provides a much clearer and more objective view of producer performance, using the same data. In other domains, the literature also reports that DEA has "outperformed" both regression and performance ratio techniques (*c.f.* [SHER95] and [BANK86A]). Various models of software production have also appeared in the DEA literature. A production model specifies the inputs and outputs to be included in the DEA analysis and, thus, play an important role in the management control process. A new model of software production is presented in [REES93] and [PARA95], which builds on previous models in the literature, but considers more outputs and, therefore, more production trade-offs. This model is limited to analyzing homogenous data sets comprised of either new development projects or comparable maintenance projects. Also of note is the two stage model by [BANK91B]. Although the model is quite simplistic, the authors have recognized and considered a crucial view of software production: the sub-process level.

More work is needed in developing more sophisticated multi-stage models since different technologies and methods can be used in different sub-processes or phases of software production. Furthermore, many real-life software projects are comprised of both new and modified code. As has been widely recognized in the software literature, work on new code development and modification of existing code are very different activities (*c.f.* [JONE91]). DEA software production models should recognize and account for the individual attributes of these two different activities in order that they may be of greater utility to practitioners.

Although DEA was originally intended to assess *operational efficiency*, recent research has extended DEA to consider the attainment of producer goals and, hence, assess *effectiveness*. Measuring the effectiveness of an organization in meeting its objectives is of considerable importance since it is fundamental to (the evaluation component of) the management control function (*c.f.* [ANTH89]). In order to move towards measuring effectiveness and recognize these goals, DEA multipliers must reflect realistic values or prices which are often imprecisely known, being represented by an upper and lower bound. To accomplish this, various models and approaches for restricting DEA multipliers have been presented in the DEA literature (such as [CHAR90]). However, the exact relationship between rigorously defined traditional economic effectiveness measures, such as measures of overall efficiency (see [FÄRE88]), and multiplier flexibility has not been fully investigated in the literature. In order to properly apply DEA multiplier restrictions to measure effectiveness, one must be able to define and understand what is being measured. Thus, the relationship between these methods must be further understood.

Software project forecasting and estimation is crucial to software project management since poor forecasts can be the root cause of many schedule and cost overruns or even project failures (*c.f.* [HUMP89], [PRES92] and [ABDE91]). Thus, forecasting and estimation are fundamental to management planning and form the basis for the software project plan. This plan defines the tasks to be completed by the project team, and provides a framework for project management and control. During the course of many software projects, changes to the project scope occur or other problems and crises arise that necessitate constant revisions to the project objectives. However, it is not always possible to change one objective without affecting one or more of the others. One must then attempt to balance the priorities and objectives as best as possible, to accommodate the changes to the project plan, by means of tradeoff analysis.

Rosen *et al.* [ROSE95] present a general framework and methods for the computation of marginal rates - a special case of tradeoffs. However, marginal rates are limited to assessing the impact of infinitesimal (or in the case of DEA, very small finite) changes to one or more variables while project managers are interested in more general or larger than single unit tradeoffs. Although these methods provide management with very useful information, they are not well suited for the general tradeoff analysis required for ongoing project planning and management.

1.3. Thesis Objectives

This thesis will attempt to address several limitations of DEA techniques that arise through their application to software engineering management. These limitations are discussed in Section 1.2. As a result, the thesis will have several related objectives. The *first* objective will be to propose several multi-stage models of software production. These models would allow analysts to "drill down" to the subprocess level, and evaluate performance at different phases of production. Furthermore, these models must be applicable for evaluating data sets containing projects with varying degrees of new and modified code.

Measuring effectiveness is fundamental to the management control process. This is the motivation for the *second* objective: researching the relationship between traditional economic production measures (and definitions) and multiplier flexibility. This will result in a prescriptive framework for the application of DEA models to measure overall efficiency and effectiveness, and the creation of several new DEA models important to this framework.

The *final* objective is related to the application of DEA to software project planning, specifically: forecasting and tradeoff analysis. Existing techniques presented by Rosen *et al.* will be adapted for the purpose of conducting general tradeoff analysis. Inherent in this analysis process is the generation of efficient project forecasts.

1.4. Outline

The outline of the thesis proposal is as follows:

- Chapter 2 provides an overview of Data Envelopment Analysis. Methods for restricting DEA multiplier flexibility are also reviewed, as well as models based on traditional economic techniques for measuring allocative and overall efficiency.

- Chapter 3 describes existing software production models. This is followed by the presentation of several new multi-stage models of software production.

- Chapter 4 examines the relationship between overall efficiency and multiplier flexibility. In doing so, a prescriptive framework for applying DEA to measure overall efficiency and effectiveness is developed. Several new DEA models are presented to accompany the framework.

- Chapter 5 presents several new DEA methods for conducting interactive tradeoff analysis.

- Chapter 6 summarizes the thesis and lists the theoretical contributions.

CHAPTER 2 *Data Envelopment Analysis*

"We believe that DEA provides a new approach to organizing and analyzing data - 'discerning new truth' "

A. Charnes et al. [CHAR94]

2.1. The Measurement of Productive Efficiency and DEA[1]

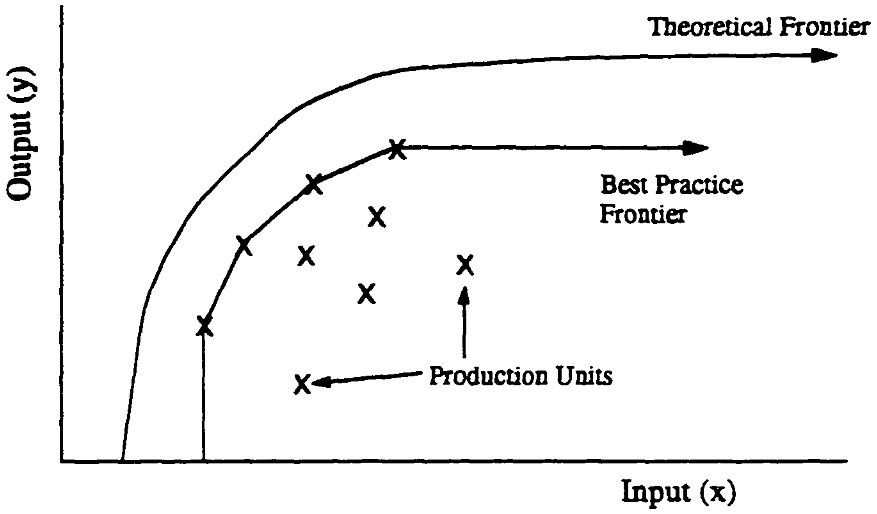
In a *production process*, firms or production units, turn resources into outcomes. Economists use the term *productive efficiency* to describe how well an organization or production unit performs in terms of utilizing its resources to generate outcomes. The relationship between the resources consumed by the production process and the resulting outputs is formally described by a *production function*. A production function gives the maximum amount of outputs that can be obtained with a certain combination of inputs. Similarly, it can describe the minimum amount of inputs required to obtain a certain level of outputs. Hence, the production function constitutes a boundary for the production possibility set, and it is also referred to as a *production frontier*.

Measurement of the efficiency of a production unit, in general, can be made relative to this production frontier. However, in practice, only observational data is available since the production frontier is seldom known. Consequently, one can

[1] The following introduction draws heavily from [ALI93], [LOVE93] and [REES93].

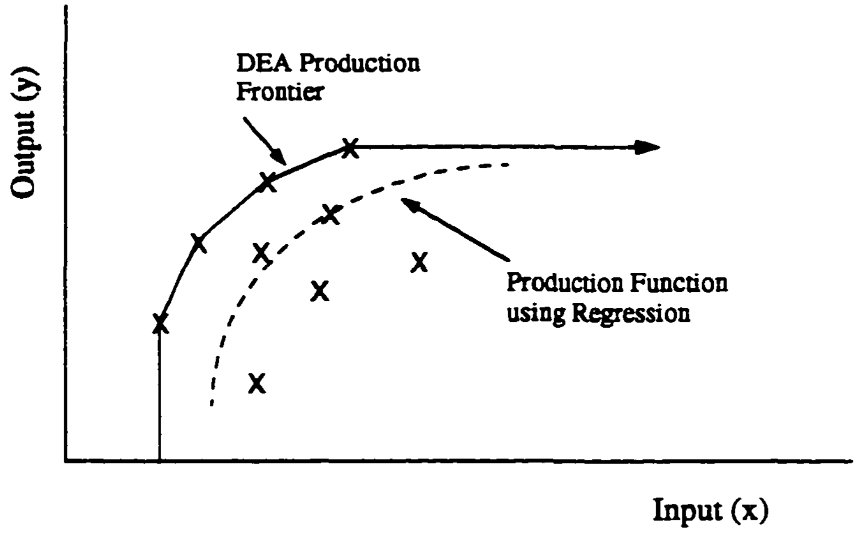
only construct an *empirical production frontier*, or *envelopment surface* against which productive efficiency can be computed. See Figure 2-1.

Figure 2-1 Theoretical Versus Best Practice Frontiers



Overall productive efficiency can be decomposed into two components: technical and allocative. *Technical efficiency* refers to either the ability of a production unit to produce as much output as input usage allows, or to the ability to use as little input as is required by output production, or some combination of the two. Hence, it deals solely with the "operational performance" of the unit and is independent of the behavioural goals of the producer. *Allocative efficiency* refers to the ability of the unit to combine inputs and outputs in optimal proportions that satisfy the behavioural objectives of the producer. These objectives include cost minimization, revenue or profit maximization or other objectives the producer pursues.

Figure 2-2 A Comparison of DEA and Regression



There are basically two approaches to the measurement of productive efficiency, each of which has its advantages and disadvantages (see Figure 2-2). The *Econometric* methods are parametric and predominantly stochastic. Parametric methods require the *a priori* specification of the functional form of the production frontier, an error distribution and sometimes and inefficiency distribution. These methods generally employ various *regression* techniques and often focus on central tendency. This hypothesized functional form cannot be directly tested, and if misspecified, the effects can be confounded with inefficiency [LOVE93]. This problem can be avoided with the use of the second approach, *mathematical programming* methods referred to as *Data Envelopment Analysis* (DEA). These methods are non-parametric and do not require assumptions about the functional form of the frontier. A weakness of these methods is that measurement error can be confused with inefficiency. However, some research efforts have attempted to incorporate stochastic features into DEA frontiers (*c.f.* [SENG89] and [BANK91A]).

2.2. DEA Models

DEA, was originally introduced by Charnes, Cooper, and Rhodes [CHAR78]. Their work builds on the seminal paper on productive efficiency by Farrell [FARR57], and extends the engineering ratio approach to efficiency measurement to multiple input-output combinations. We will briefly introduce this model, referred to as the CCR model, as well as some important variations. For reviews of the basic DEA models, the reader is further referred to [ALI93] and [CHAR94].

We consider a set of n production units, also known as *Decision Making Units* (DMUs), each consuming varying amounts of m inputs to produce s outputs. Let $\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})^T$ and $\mathbf{y}_j = (y_{1j}, \ldots, y_{sj})^T$ represent the input and output vector, respectively, for DMU_j, j = 1, ... ,n. We also employ $\mathbf{X}$ to denote the $m \times n$ matrix of inputs, and $\mathbf{Y}$ to denote the $s \times n$ matrix of outputs. Following [BANK84], the production possibility set, T is defined as:

$$T = \{ (\mathbf{x}, \mathbf{y}) \mid \mathbf{y} \text{ can be produced from } \mathbf{x} \} \qquad \text{(EQ 2-1)}$$

The production possibility set, T is constructed from the observational data $(\mathbf{x}_j, \mathbf{y}_j), j = 1, \ldots, n$ by postulating the following properties:

Convexity: If $(\mathbf{x}_j, \mathbf{y}_j) \in T$ and $\lambda_j \geq 0$ are non-negative scalars such that $\sum_{j=1}^{n} \lambda_j = 1$, then $(\sum_{j=1}^{n} \lambda_j \mathbf{x}_j, \sum_{j=1}^{n} \lambda_j \mathbf{y}_j) \in T$.

Inefficiency: a) If $(\mathbf{x}, \mathbf{y}) \in T$ and $\bar{\mathbf{x}} \geq \mathbf{x}$, then $(\bar{\mathbf{x}}, \mathbf{y}) \in T$, b) If $(\mathbf{x}, \mathbf{y}) \in T$ and $\bar{\mathbf{y}} \leq \mathbf{y}$, then $(\mathbf{x}, \bar{\mathbf{y}}) \in T$.

Ray Unboundedness: If $(\mathbf{x}, \mathbf{y}) \in T$ then $(k\mathbf{x}, k\mathbf{y}) \in T$ for any $k > 0$.

Minimum Extrapolation: T is the intersection set of all $\hat{T}$ satisfying the postulates of *convexity*, *ray unboundedness* and *inefficiency*, and subject to the conditions that each of the observed vectors $(\mathbf{x}_j, \mathbf{y}_j) \in \hat{T}, j = 1, \ldots, n$.

The *technical efficiency* of a particular DMU_0, can be evaluated with the following formulation:

$$\begin{aligned} &\text{Min} \quad \theta \\ &\text{s.t.} \quad (\theta \mathbf{x}_0, \mathbf{y}_0) \in T \end{aligned} \qquad \text{(EQ 2-2)}$$

Formulation (EQ 2-2) in turn

is equivalent to the following linear program:[1]

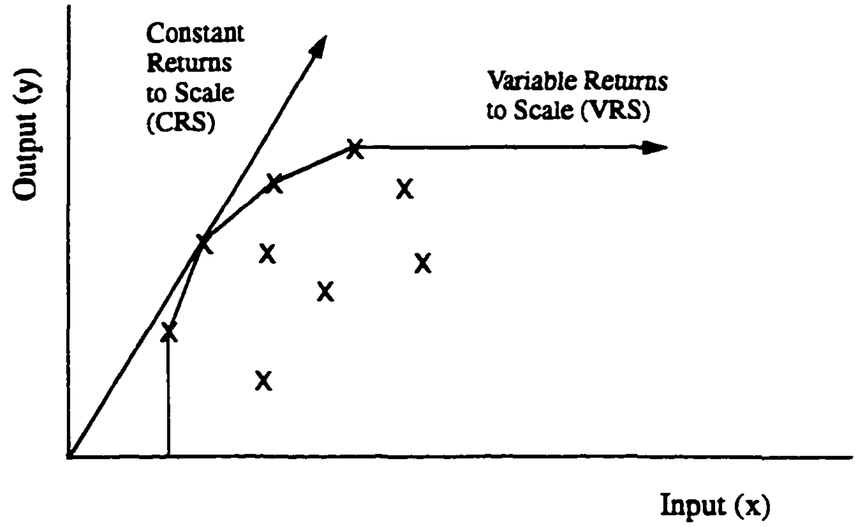
$$\begin{aligned} &\underset{\theta,\lambda}{\text{Min}} \quad \theta \\ &\text{s.t.} \qquad \mathbf{Y\lambda} \geq \mathbf{y}_0 \\ &\quad \theta\mathbf{x}_0 - \mathbf{X\lambda} \geq \mathbf{0} \\ &\quad \theta \text{ free}, \mathbf{\lambda} \geq \mathbf{0} \end{aligned} \qquad \text{(EQ 2-3)}$$

The solution to equation (EQ 2-3) gives the efficiency of DMU_0. This optimal θ is an *input-oriented, radial* measure of technical efficiency. Hence, this measure describes how efficiently resources have been utilized to produce y_0. When $\theta < 1$, DMU_0 is inefficient. In this case the DMU can proportionally decrease its inputs by θ and still produce y_0. The positive elements of the vector λ indicate the *peer group* or *reference set* of efficient DMUs located on the frontier, against which DMU_0 is evaluated. When $\theta = 1$, DMU_0 is a boundary point of the production possibility set. A boundary point may not be efficient since $(\mathbf{x}_0, \mathbf{y}_0)$ may contain slacks in any of its $m + s$ dimensions. A DMU is technically efficient *if and only if* $\{\theta = 1, \mathbf{X\lambda} = \mathbf{x}_0, \mathbf{Y\lambda} = \mathbf{y}_0\}$.

[1] Following **[SEIF90]**, we do not employ models with non-archimedian infinitesimals and slacks in the objective function to simplify this exposition. For DEA computational methods and algorithms, the reader is referred to **[ALI94]**.

The CCR envelopment surface consists of hyperplanes that form particular facets of the *conical* hull of points $(\mathbf{x}_j, \mathbf{y}_j), j = 1, \ldots, n$ and exhibits *constant returns to scale* (**CRS**). Accordingly, proportionate increases in inputs result in proportionate increases in outputs, while keeping the input and output mix for $(\mathbf{x}_0, \mathbf{y}_0)$ constant ([BANK92]). The frontier exhibits CRS (as opposed to variable returns to scale or VRS) due to the *ray unboundedness* postulate (see Figure 2-3).

Figure 2-3 CRS and VRS Frontiers



The dual linear program of (EQ 2-3) is as follows:

$$\max_{\mu, v, \omega} \quad \mathbf{y}_0^T \mu \qquad \text{(EQ 2-4)}$$

$$\text{s.t.} \quad \mathbf{x}_0^T \mathbf{v} = 1$$
$$\mathbf{Y}^T \mu - \mathbf{X}^T \mathbf{v} \le \mathbf{0}$$
$$\mu, \mathbf{v} \ge \mathbf{0}$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_m)^T$ and $\mu = (\mu_1, \mu_2, \ldots, \mu_s)^T$ are vectors of input and output weights (or multipliers). Efficiency is measured as a function of these weights. Each DMU is then 'assigned' by the linear program the weights which maximize its efficiency, provided that the set of weights yields efficiency scores that do not exceed unity, for all DMUs.

An important variation of this original model was developed by Banker, Charnes and Cooper [BANK84]. This model, designated the BCC model, can be derived from the previous postulates by removing the *ray unboundedness* postulate. The resulting linear program has the following form, where $\mathbf{1}^T = (1, ..., 1)$:

$$\begin{aligned} &\min_{\theta, \lambda} \quad \theta \\ &\text{s.t.} \quad \mathbf{Y}\boldsymbol{\lambda} \geq \mathbf{y}_0 \\ &\theta \mathbf{x}_0 - \mathbf{X}\boldsymbol{\lambda} \geq \mathbf{0} \\ &\mathbf{1}^T\boldsymbol{\lambda} = 1 \\ &\theta \text{ free}, \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \qquad \text{(EQ 2-5)}$$

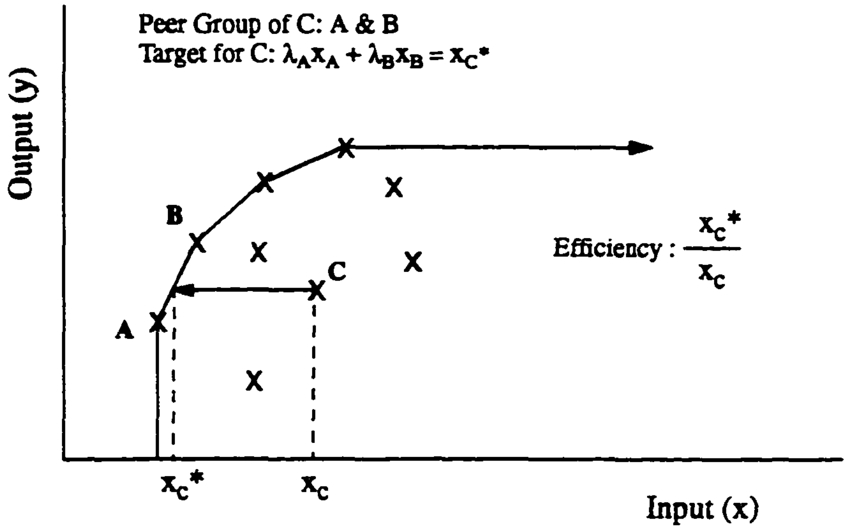
The BCC envelopment surface exhibits *variable returns to scale* (**VRS**) as depicted in Figure 2-3, and consists of hyperplanes that form particular facets of the *convex* hull of the points $(\mathbf{x}_j, \mathbf{y}_j), j = 1, ..., n$. Regions of a VRS frontier may exhibit *increasing*, *constant* or *decreasing* returns to scale. See Figure 2-4 for a simple illustration of this model.

The dual linear program of (EQ 2-5) is as follows:

$$\begin{aligned} &\max_{\mu, \nu, \omega} \quad \mathbf{y}_0^T\boldsymbol{\mu} + \omega \\ &\text{s.t.} \quad \mathbf{x}_0^T\boldsymbol{\nu} = 1 \\ &\mathbf{Y}^T\boldsymbol{\mu} - \mathbf{X}^T\boldsymbol{\nu} + \omega\mathbf{1} \leq \mathbf{0} \\ &\boldsymbol{\mu}, \boldsymbol{\nu} \geq \mathbf{0} \end{aligned} \qquad \text{(EQ 2-6)}$$

where ω is a measure of scale efficiency [BANK84].

Figure 2-4 An Input-Oriented VRS (BCC) Example


Peer Group of C: A & B
Target for C: $\lambda_A x_A + \lambda_B x_B = x_C^*$
Output (y)
B
C
A
Efficiency : $\frac{x_C^*}{x_C}$
x_C^*
x_C
Input (x)

Other measures of technical efficiency also exist. *Output-oriented* radial technical efficiency is measured in the following manner:

$$\text{Max} \quad \phi \qquad \text{(EQ 2-7)}$$

$$\text{s.t.} \quad (\mathbf{x}_0, \phi\mathbf{y}_0) \in T.$$

where T is the production possibility set, consisting of a VRS, CRS or other form of frontier or production technology (exhibiting other returns to scale or RTS).

2.2.1. Extensions to the Basic DEA Models

The basic DEA models described in the previous section can be characterized by the geometry of the envelopment surface (and their inherent RTS properties) as well as the manner of projection to the efficient frontier. The many variations of both frontier and projection allow for considerable power and flexibility in the application of DEA to a great number of problem domains. Moreover, a number of powerful extensions to these basic models can be found in the DEA literature; a thorough review of these extensions, the reader is referred to [CHAR94]. Furthermore, Section 2.3. reviews another important extension and generalization of the basic DEA models: the cone-ratio models.

One important extension is the nondiscretionary variable first introduced by Banker and Morey [BANK86B]. The models presented in the previous section implicitly assume that all variables (inputs and outputs) are discretionary and are fully controlled and varied by the DMU and its management. However, in many applications there exists inputs and outputs that are exogenous or nondiscretionary and are beyond the control of management. Let D represent the subset of discretionary variables and N the subset of nondiscretionary variables. Furthermore,

$$I = \{1, 2, \dots, m\} = I_D \cup I_N, I_D \cap I_N = \varnothing, \qquad \text{(EQ 2-8)}$$
$$O = \{1, 2, \dots, s\} = O_D \cup O_N, O_D \cap O_N = \varnothing,$$

where I and O represent the set of input and output variables. The model formulation for the nondiscretionary input-oriented BCC model is given by:

$$\begin{aligned} &\min_{\theta, \lambda} \ \theta && \text{(EQ 2-9)} \\ &\text{s.t.} \quad \mathbf{Y\lambda} \ge \mathbf{y}_0 \\ &\quad \theta \mathbf{x}_{i0} - \mathbf{X}_i \mathbf{\lambda} \ge \mathbf{0}, \quad i \in I_D \\ &\quad -\mathbf{X}_j \mathbf{\lambda} \ge -\mathbf{x}_{j0}, \quad j \notin I_D \\ &\quad \mathbf{1}^T \mathbf{\lambda} = 1 \\ &\quad \theta \text{ free}, \mathbf{\lambda} \ge \mathbf{0} \end{aligned}$$

where $\mathbf{x}_i, i \in I_D$ represents a vector of the subset of discretionary inputs, $\mathbf{x}_j, j \notin I_D$ represents a vector of the subset of nondiscretionary inputs, and $\mathbf{X}_i$ and $\mathbf{X}_j$ a matrix containing vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively.

Note that θ minimizes only the discretionary inputs $\mathbf{x}_i, i \in I_D$. In contrast, the nondiscretionary inputs $\mathbf{x}_j, j \notin I_D$ have only an indirect effect on the efficiency score since these input levels are beyond managerial control.

While the nondiscretionary variable extension was presented in terms of the BCC input-oriented model, this extension is applicable to all other basic models with the appropriate modifications.

2.3. Methods of Bounding DEA Weights

One of the most comprehensive approaches to bounding DEA multipliers suggested in the literature, is the Cone Ratio model (see [CHAR89], [CHAR90]). This approach generalizes the standard DEA models of Section 2.2. by requiring that input and output weights be restricted to given closed cones. These multiplier bounds are typically established from market prices or managerial judgement (see APPENDIX C for some practical examples). Such restrictions can be introduced into DEA models by incorporating additional inequality constraints, generally of the following form, into the multiplier problems:

$$\mathbf{A}^o\mu + \mathbf{A}^i\mathbf{v} \geq \mathbf{0} \qquad \text{(EQ 2-10)}$$

where $\mathbf{A}^i$ and $\mathbf{A}^o$ are $(k \times m)$ and $(k \times s)$ matrices respectively, and k is the total number of multiplier constraints; or, equivalently:

$$\begin{aligned} \mathbf{w} \in W &= \{\mathbf{w}: \mathbf{A}\mathbf{w} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0}\}, \\ \mathbf{A} &= [\mathbf{A}^o, \mathbf{A}^i], \mathbf{w}^T = (\mu^T, \mathbf{v}^T). \end{aligned} \qquad \text{(EQ 2-11)}$$

When $\mathbf{A}^o = \mathbf{0}$ these constraints are collectively referred to as an *input* cone where $\mathbf{v} \in V = \{\mathbf{v}: \mathbf{A}^i\mathbf{v} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}\}$ (see Figure 2-5); alternately when $\mathbf{A}^i = \mathbf{0}$ the constraints are called an *output* cone where $\mu \in U = \{\mu: \mathbf{A}^o\mu \geq \mathbf{0}, \mu \geq \mathbf{0}\}$. The constraints may also constitute two separate input and output cones: together called a *separable* cone. However, if at least one of the constraints relate an input and an output multiplier then (EQ 2-10) is known as a *linked* cone.

The previous cones are known as *intersection form* cones since they are defined by the intersection of a finite number of half-spaces. These cones can be equivalently represented in the *sum* form, spanned by a finite number of extreme vectors $\mathbf{b}_i$:

$$W = \{\mathbf{B}^T\alpha : \alpha \geq \mathbf{0}\}, \alpha \in E_+^l \qquad \text{(EQ 2-12)}$$

$$\mathbf{B}^T = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_l), \mathbf{b}_i \in E_+^{m+s} \quad \forall i$$

and:

$$\mathbf{w} \in -W^* = \{\mathbf{w} : \mathbf{B}\mathbf{w} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0}\}, \qquad \text{(EQ 2-13)}$$

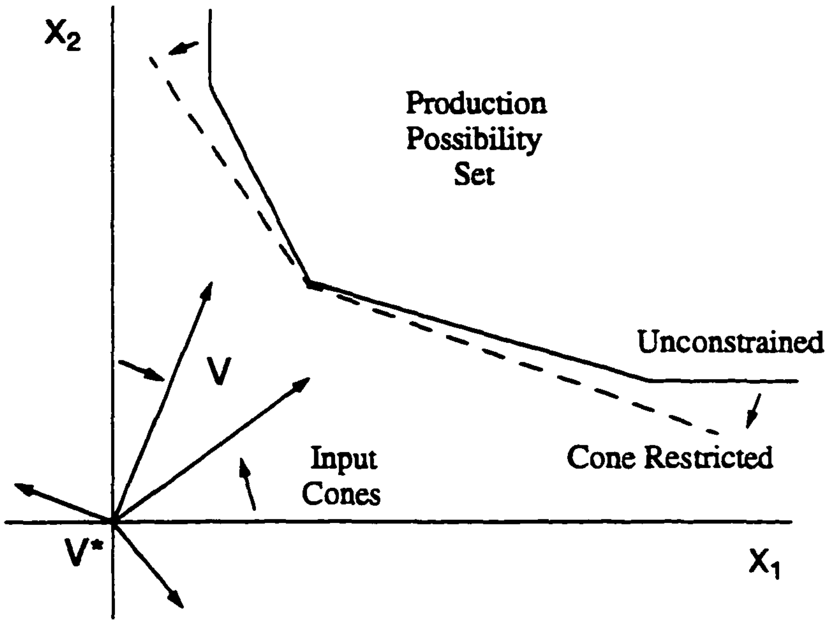
$$\mathbf{B} = [\mathbf{B}^o, \mathbf{B}^i], \mathbf{w}^T = (\mu^T, \nu^T).$$

where W^* is the negative polar cone of W. Accordingly, the spanning vectors of W^* are the normal directions of the hyperplanes bounding W (provided that W is polyhedral). See Figure 2-5 for an example of an input cone and its polar cone.

The cones U and V, in sum form, can be incorporated directly into standard DEA models by transforming the observed production data. Letting $\bar{\mathbf{X}} = \mathbf{B}^i\mathbf{X}$ and $\bar{\mathbf{Y}} = \mathbf{B}^o\mathbf{Y}$ a cone-ratio model corresponds to a standard DEA model, such as the CCR or BCC model, but with the transformed data $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. The sum form cone ratio model, as presented in [CHAR90], can utilize only separable, input and output cones. This model must be extended or generalized in order to incorporate linked sum form cones as well; however, this issue has not yet been addressed in the DEA literature.

A sum form constraint matrix $\mathbf{B}$ can be written in intersection form as $\mathbf{A} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}$, where $\mathbf{B}^T$ is of full column rank [CHAR90]. Similarly, an intersection form constraint matrix $\mathbf{A}$ can be written in sum form as $\mathbf{B}^T = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$, where **A** is full column rank. See [YU85] and [TAMU76] for a discussion of issues associated with these transformations.

Figure 2-5 An Input Cone Example



A series of approaches have been suggested to establish the multiplier restrictions of (EQ 2-10). For reviews of these methods, the reader is referred to [CHAR94], [ALI93] and [ROLL93]; for applications *c.f.* [THOM90] and [CHAR90].

One such approach is based on the 'Assurance Region' concept, which also restricts *relative* multiplier values (*c.f.* [THOM90], [THOM92]). A *single* input weight is used as a basis for comparison against *all other* input weights. Similarly, weights of all outputs are compared to a single output weight. A typical set of constraints could be:

$a_i\mu_1 \le \mu_i \le b_i\mu_1$ for all i; (EQ 2-14)

$a_i\nu_1 \le \nu_i \le b_i\nu_1$ for all j;

where a_i and b_i are lower and upper bounds, and μ_1 and ν_1 are the numeraires. Similarly, a series of pairwise comparisons can be made between the multipliers. A typical set of constraints for this type of assurance region could be:

$$a_{ij} \le \mu_i / \mu_j \le b_{ij} \; ; \qquad \text{(EQ 2-15)}$$

$$a_{ij} \le \nu_i / \nu_j \le b_{ij} \; ;$$

$$\text{where } i \ne j, \text{ with } 0 \le a_{ij} \le b_{ij}.$$

These constraints can also be represented as in (EQ 2-10). A more restrictive situation is where the input weights are tied to output weights. Hence, the input and output cones are 'linked'. However, linking input and output cones can have some 'side effects' in some situations. For example, standard CCR models with linked price cones make an implicit assumption of 'zero maximum profit' which may be too restrictive for some situations (*c.f.* [**THOM92**]).

Wong and Beasley [**WONG90**] present a method to place limits on the proportion of total (virtual) output of DMU$_k$ of which a particular output i comprises. This can be represented as follows:

$$a_i \le \; \mu_i y_{ik} \; / \; \left(\sum\nolimits_{r=1}^{s} \mu_r y_{rk}\right) \; \le b_i \qquad \text{(EQ 2-16)}$$

Value judgements are used to determine the levels of a_i and b_i. Paradi, Reese and Rosen [**PARA95**] present a similar approach whereby limits are set on the contribution of particular performance ratio to an overall 'z-score' (a weighted sum of performance ratios). Value judgements obtained from management are used to obtain bounds of the following form:

$$a_j \le \; W_j Z_{jk} / \left(\sum\nolimits_i W_i Z_{ik}\right) \; \le b_j \qquad \text{(EQ 2-17)}$$

where Z_{jk} is a performance ratio value, and W_j (non-DEA weight) is a weight representing ratio importance. The limits a_j and b_j are used to determine DEA multiplier bounds. Paradi *et al.* further show how (EQ 2-17), and hence, (EQ 2-16), can be transformed to the form of (EQ 2-10) and they further give different constraint sets that can be derived from the bounds of (EQ 2-17).

Other miscellaneous approaches have been presented in the literature, which impose relationships on multipliers other than individual weight bounds. Ali *et al.* **[ALI91]** present models for ordinal relations between DEA multipliers. For example, it may be desirable to constrain μ_1 through the following ordinal relationship: $\mu_1 \geq \mu_2 + \mu_3$. Other researchers have applied such ordinal relations in conjunction with data normalization or other data preparation techniques (*c.f.* **[KORN91]**, **[ROLL91]**).

Roll, Cook and Golany argue that when information is not available regarding the relative importance of the different model factors, the weight variation within each factor can still be restricted **[ROLL91]**. They suggest running an unbounded model, examining the variation of the factor weights, and then restricting the amount of weight variation to *d*:1 - a ratio of the highest weight value to the lowest weight value. The basic DEA models can then be extended by adding constraints of the type:

$$\frac{2 \times w_i}{1+d} \leq w_{ij} \leq \frac{2 \times d \times w_i}{1+d}, \quad \forall j \qquad \text{(EQ 2-18)}$$

where w_i is an input or output weight, and d is the ratio constraining allowable variation. Variations of this basic approach are discussed in **[ROLL91]** and **[ROLL93]**. Other techniques that involve weight bounding procedures include utilizing a constant set of weights (*c.f.* **[ROLL91]**), as well as cross-efficiency studies (*c.f.* **[DOYL94]**).

2.4. Measures of Overall Efficiency

Defining and measuring overall efficiency requires that a *behavioural objective* be specified, along with value or pricing information. Traditionally, these objectives have been revenue maximization, cost minimization or profit maximization. However, alternate behavioural objectives have been presented in the literature, as well as constraints that impede the achievement of behavioural goals. These constraints could be regulatory or some other form of non-technological constraint.

For some comprehensive works that include the measurement of overall efficiency using DEA techniques, the reader is further referred to **[FÄRE86]**, **[FÄRE88]**, and **[FÄRE94]**, as well as the references contained therein; for DEA models and applications *c.f.* **[BANK88]**, **[BANK93]**, **[FERR94]**, and **[GOLA93]**; for DEA computational algorithms refer to **[BARR97]**, **[ALI94]** and **[SUEY92]**; for reviews of stochastic allocative methods refer to **[RETZ92]**.

If the objective of the production unit(s), or the objective assigned by the analyst, is cost minimization, then the input prices $\mathbf{c} \geq \mathbf{0}$ must be known. The overall *minimum cost* of producing output vector y_0 is obtained by solving the following:

$$\begin{aligned} \min_{x,\lambda} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{Y\lambda} \geq \mathbf{y}_0 \\ & \mathbf{x} \geq \mathbf{X\lambda} \\ & \mathbf{1}^T\mathbf{\lambda} = 1 \\ & \mathbf{\lambda} \geq \mathbf{0} \end{aligned} \tag{EQ 2-19}$$

for a VRS frontier. Overall *cost efficiency* ($\mathrm{OE_i}$) is determined by dividing overall minimum cost $\mathbf{c}^T\mathbf{x}^*$ by observed cost:

$$OE_i = \frac{\mathbf{c}^T\mathbf{x}^*}{\mathbf{c}^T\mathbf{x}_0} \tag{EQ 2-20}$$

where $0 \le OE_i \le 1$. Overall cost efficiency can be decomposed into two component measures: input radial technical efficiency ($\mathbf{TE_i}$) and input allocative efficiency ($\mathbf{AE_i}$), where:

$$OE_i = TE_i \cdot AE_i \tag{EQ 2-21}$$

Once technical efficiency is obtained by solving a model such as (EQ 2-5), allocative efficiency can be derived from (EQ 2-21).

Alternatively, if the objective of the production unit(s) is known to be, or assumed to be, revenue maximization, then actual revenue can be calculated for each observation provided the output prices $\mathbf{r} \ge \mathbf{0}$ are known. The overall maximum revenue for input vector $\mathbf{x}_0$ is obtained from the following:

$$\begin{aligned} \underset{y,\lambda}{\text{Max}} \quad & \mathbf{r}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{Y\lambda} \ge \mathbf{y} \\ & \mathbf{x}_0 \ge \mathbf{X\lambda} \\ & \mathbf{1}^T\mathbf{\lambda} = 1 \\ & \mathbf{\lambda} \ge \mathbf{0} \end{aligned} \tag{EQ 2-22}$$

assuming a VRS frontier. Overall *revenue efficiency* is defined as:

$$OE_o = \frac{\mathbf{r}^T\mathbf{y}^*}{\mathbf{r}^T\mathbf{y}_0} \tag{EQ 2-23}$$

where $OE_o \ge 1$. Once output technical efficiency ($\mathbf{TE_0}$) has been obtained, output allocative efficiency ($\mathbf{AE_0}$) can be derived from the following relationship:

$$OE_o = TE_o \cdot AE_o. \tag{EQ 2-24}$$

Thus far, we have only considered VRS production frontiers. Under CRS frontiers, maximum profit is implicitly zero [FÄRE88]. Thus, $\mathbf{r}^T\mathbf{y}^* = \mathbf{c}^T\mathbf{x}^*$ and the following result is obtained for CRS models:

$$\frac{OE_o}{OE_i} = \frac{\mathbf{r}^T\mathbf{y}_0}{\mathbf{c}^T\mathbf{x}_0}. \qquad \text{(EQ 2-25)}$$

In both the overall cost and revenue efficiency models, the decomposition into technical and allocative efficiency provides insight into the relationship between these models and the standard DEA models introduced in Section 2.2. Notice how both the measure of technical efficiency, and the pricing information used to assess allocative efficiency, must be consistent with the behavioural objective in order to measure overall efficiency.

The non-parametric economic production analysis literature[1] presents other measures of overall efficiency (*c.f.* [BANK88] and [CHAV94]). Extending the work of Varian [VARI84], Banker and Maindiratta present a class A of non-parametric production sets and well-defined measures that can be used to assess overall profit efficiency [BANK88]. The class A includes all closed, convex and monotone (sometimes referred to as free disposability) production possibility sets constructed from observational data $(\mathbf{x}_j, \mathbf{y}_j), j = 1, \ldots, n$ that have at least one DMU which is consistent with profit maximization relative to all production possibilities, for the observed revenue and cost data $\mathbf{p}_j = (\mathbf{r}_j, \mathbf{c}_j), j = 1, \ldots, n$ (see [BANK88] for a formal definition). The authors show that class A is bounded by the following outer bound L and inner bound S:

$$L = \{ (\mathbf{y}, -\mathbf{x}) \mid \mathbf{r}_j\mathbf{y} - \mathbf{c}_j\mathbf{x} \le \mathbf{r}_j\mathbf{y}_j - \mathbf{c}_j\mathbf{x}_j \quad \forall j \in E \ , \mathbf{y} \ge \mathbf{0}, \mathbf{x} \ge \mathbf{0} \} \qquad \text{(EQ 2-26)}$$

[1] This literature includes non-parametric models not typically associated with DEA literature - in general, those models not included in [CHAR94].

$$S = \{ (\mathbf{y}, -\mathbf{x}) \mid \mathbf{y} \le \sum_{j=1}^{n} \lambda_j \mathbf{y}_j , \mathbf{x} \ge \sum_{j=1}^{n} \lambda_j \mathbf{x}_j , \sum_{j=1}^{n} \lambda_j = 1, \quad \text{(EQ 2-27)}$$
$$\mathbf{y} \ge \mathbf{0}, \mathbf{x} \ge \mathbf{0} \}$$

where E is the set of all DMUs in the observed data that maximize profit for the observed prices, relative to all other DMUs. Notice that S corresponds to the BCC (VRS) production possibility set. This latter result established an important link between the previously separate DEA and non-parametric production analysis bodies of literature [BANK88].

As presented in Banker and Maindiratta [BANK88], an overall profit efficiency measure for DMU$_j$ with an actual profit $\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j > 0$ can be defined as:

$$min_{\mathbf{x},\mathbf{y}} \{ (\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j) / (\mathbf{r}_j^T\mathbf{y} - \mathbf{c}_j^T\mathbf{x}) \mid (\mathbf{y}, -\mathbf{x}) \in T, \mathbf{r}_j^T\mathbf{y} - \mathbf{c}_j^T\mathbf{x} \ge 0 \} . \quad \text{(EQ 2-28)}$$

A corresponding technical efficiency measure of DMU$_j$ with an actual profit $\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j > 0$ can be defined as:

$$min_h \{ (\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j) / (\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T h\mathbf{x}_j) \mid (\mathbf{y}, -\mathbf{x}) \in T, \mathbf{r}_j^T\mathbf{y} - \mathbf{c}_j^T\mathbf{x} \ge 0 \} . \quad \text{(EQ 2-29)}$$

In both of the definitions (EQ 2-28) and (EQ 2-29), $T \in A, S \subseteq T \subseteq L$. Allocative efficiency can obtained by dividing overall profit efficiency by technical efficiency. Note that (EQ 2-28) is not suitable for CRS frontiers since maximum profit is zero in this case, as stated earlier. Furthermore, the overall maximum profit $\mathbf{r}_j^T\mathbf{y} - \mathbf{c}_j^T\mathbf{x}$ must be greater than zero for (EQ 2-28) to be meaningful.

Notice that both measures, based on profit ratios, are not useful in situations where DMU$_j$ has an actual profit $\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j \le 0$. Furthermore, in the overall profit efficiency measure, both the inputs and outputs are reduced when obtaining the efficient projection (in the denominator) whereas the inputs only are reduced in the technical efficiency measure. This is inconsistent with the approach taken by Farrel [FARR57] upon which (EQ 2-19) to (EQ 2-24) are based. Finally, the technical efficiency measure contains prices and thus considers the implicit behavioural goal of profit maximization. This is in sharp contrast to the approach of Farrell where technical efficiency is measured independent of prices and behavioural goals.

The non-parametric production analysis literature contains additional efficiency measures that incorporate both quantities and prices for the inputs and outputs. For instance, Chavas and Cox [CHAV94] apply the following linear programs to obtain input and output distance functions $D_L(\mathbf{x}_j, \mathbf{y}_j)$ and $F_L(\mathbf{x}_j, \mathbf{y}_j)$ respectively:

$$1/D_L(\mathbf{x}_j, \mathbf{y}_j) = min_\delta[\delta : \mathbf{r}_i{}^T\mathbf{y}_j - \mathbf{c}_i{}^T\mathbf{x}_j\delta \le \mathbf{r}_i{}^T\mathbf{y}_i - \mathbf{c}_i{}^T\mathbf{x}_i, i \in E], \quad \text{(EQ 2-30)}$$

$$1/F_L(\mathbf{x}_j, \mathbf{y}_j) = max_\delta[\delta : \mathbf{r}_i{}^T\mathbf{y}_j\delta - \mathbf{c}_i{}^T\mathbf{x}_j \le \mathbf{r}_i{}^T\mathbf{y}_i - \mathbf{c}_i{}^T\mathbf{x}_i, i \in E]. \quad \text{(EQ 2-31)}$$

Both are based on the upper bound production possibility set L. These distance functions are often employed to construct sequential and intertemporal productivity indices (see [CHAV94] and [GROS93]).

2.5. Summary

Measures of *productive efficiency* describe how well organizations or production units are utilizing inputs to generate outcomes. Overall productive efficiency can be decomposed into two components: *technical* and *allocative* efficiency. While technical efficiency deals solely with *operational* performance, allocative efficiency considers the mix of inputs and outputs in light of the *behavioural goals* of the producer. These objectives include cost minimization, and revenue or profit maximization.

Data Envelopment Analysis (DEA) is a set of non-parametric linear programming techniques that can be used to measure productive efficiency. In this chapter, different efficiency measures as well as types of production frontiers are reviewed. Since DEA is a linear programming technique, each model has a primal, or envelopment model and an equivalent dual or multiplier formulation.

Most real-world DEA applications require bounding, or restricting, the DEA multipliers (also known as weights). Weight bounding is done for several reasons: to correct for unreasonable values of multipliers that can lead to overly optimistic measures of technical efficiency; or to ensure that multipliers reflect realistic prices or other value measures and, hence, move from measuring technical to overall efficiency. Alternate methods and models for restricting weight flexibility are reviewed herein. Many of the these techniques can be used to bound multipliers when market prices or other value measures are not known precisely. However, in many practical situations, market prices do not exist or are unavailable and one must utilize value judgements.

When pricing information is known precisely, then overall efficiency can be measured. Defining and measuring overall efficiency requires that a behavioural objective first be specified for purposes of analysis. Traditionally, these goals have been cost minimization, revenue maximization or profit maximization. Definitions and DEA models are presented to measure overall efficiency for these behavioural goals.

CHAPTER 3 *Modelling Software Production*

> *"The first step towards the management of disease was replacement of demon theories and humours theories by the germ theory. That very step, the beginning of hope, in itself dashed all hopes of magical solutions. It told workers that progress would be made stepwise, at great effort, and that a persistent unremitting care would have to be paid to a discipline of cleanliness. So it is with software engineering today."* *Frederick P. Brooks*

3.1. Background[1]

Cost and resource estimation represents a sizeable portion of the software engineering literature. *Cost estimation* models are usually the result of engineering judgement and the extensive analysis of large databases of project data. The usual approach is to formulate a parametric model, or mathematical function of several variables, and then to apply statistical techniques to the project data to reduce the number of variables and estimate their coefficients. The main goal is to examine relationships between project cost (or effort) and various project factors, for *predictive* purposes. The reader is further referred to **[BOEH81]**, **[CONT86]**, **[KEME87]**, **[FENT91]** and **[SAGE95]**, for some comprehensive reviews on cost estimation.

In contrast, when measuring *productive efficiency*, one is more interested in assessing the performance of production processes (see Section 2.1.). Thus, one is interested in identifying the main resources (inputs) and the relevant outcomes (outputs) of the process, and in finding appropriate measures for each. This speci-

[1] Section 3.1. and Section 3.2. are based on **[REES93]** and **[PARA95]**.

fication of a production model constitutes a critical step in the application of DEA to measure efficiency.

3.2. Applications of DEA to Software Production

Several applications of DEA to software production have been reported in the literature. The production models reported in these studies are summarized in Table 3-1.

Table 3-1 Review of Software Production Models from Previous DEA Studies.

Study	Inputs		Outputs			
	Labour (hrs or $)	Other Expenses	Function Points	SLOC Lines of Code	Quality	Project Duration
Banker and Kemerer, 1989	X			X		
Elam, 1991	X	X			X	
Banker, Datar and Kemerer, 1991	X		X	X		
Reese, 1993	X	X	X		X	X
Paradi, Reese and Rosen, 1994	X	X	X		X	X

Notice that all the models have used *effort*, measured either as labour hours or cost, as the main input. The main output for the models is the *size* of code delivered. Traditionally, the most common measure for this attribute was the number of source lines of code (SLOC). Jones [JONE86] discusses some of the limitations and problems associated with this measure. However function points (FP) has gained acceptance as a more reliable measure of size (*c.f.* [ALBR79], [JONE86]). Function point methods attempt to measure the underlying program functions independent of the language. Note that function point counts may contain both new and modified code. In spite of its current popularity, the function point methods have not gone without some criticism (*c.f.* [FENT94]). Other outputs used in the models are software *quality* and project *duration* measures

The simple (one-input/one-output) production model by Banker and Kemerer [BANK88] was used to estimate the most productive scale size of software development projects. Later, Banker, Dater and Kemerer [BANK91B] used both SLOC and FPs in order to study the effects of project characteristics on different phases of the software maintenance process. The maintenance process was divided into two phases: *analysis/design* and *coding/testing*. The model used by Elam [ELAM91] also considered a quality attribute of the software as output. A limitation of this study is that most of the measures were normalized: labour cost per employee; FPs per work-month; and quality was measured by the total rework hours per FP. This normalization can be undesirable, since it removes the scale (or size) component from the analysis by assuming CRS.

In [REES93] and [PARA95] the authors present a software production model with one input and three outputs (see Figure 3-1). The model was developed in conjunction with management from two large Canadian banks. Thus, the model reflects management experience at both banks and utilizes information deemed relevant, and already gathered by the banks. The single input, project cost, is a measure of effort and reflects the cost of the software project which includes labour, overhead, computer charges, and other project costs.

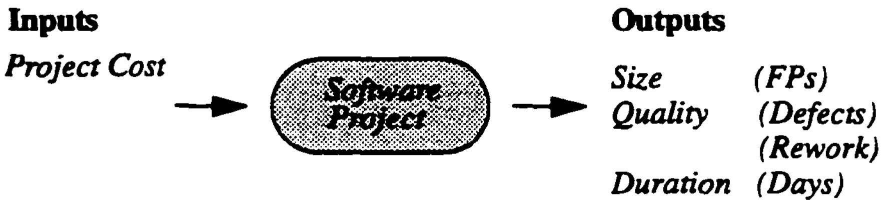


Figure 3-1 Software Production Model of Reese and Paradi *et al.*

The size of the produced software is measured as FPs. The model also includes quality as one of its outputs, as does that by Elam. However, Paradi *et al.* used data where the quality measure was different at each bank: the number of defects detected in the four month period after implementation versus rework hours from final independent testing. Both are non-ratio measures of quality as opposed to the ratio measure of quality, rework hours per FP, in **[ELAM91]**. The model further considers project duration as the final output. This output measures the calendar duration of the project (analysis, design and coding) until final independent testing, and addresses the time element of software projects and its trade-offs with the other outputs. Putnam and Meyers **[PUTN92]**, for instance, report finding interactions between time and other project measures, such as size, cost and quality. In particular, the authors found sizeable time-cost trade-offs in a large sample of software projects. Previous DEA production models have not addressed the important time component of software projects.

In order to use the quality and duration variables as outputs, some transformations are necessary on these two variables. Reese and Paradi *et al.* subtracted each observed quality and duration measure from the maximum observed respective output: $y_{ij}' = \max_j (y_{ij}) - y_{ij}$. An advantage of this transformation is that the units of the original data are maintained.

It is important to present some of the further limitations of the production model and some possible improvements that are discussed in **[PARA95]**. First, the measures used for quality (number of software defects and rework hours) correspond to a very narrow definition that misses many of the quality attributes suggested by McCall *et.al.* **[McCA77]**, Boehm *et.al.* **[BOEH78]**, Grady *et al.* **[GRAD87]**, and others. Furthermore, other significant attributes related to quality, such as customer (or user) satisfaction, should be important additions to the model.

Moreover, the model presented by Paradi *et al.* does not include environmental factors which may be relevant in the production process. Hence, the effect of these factors will be implicit in the efficiency measures. However, these factors can be easily incorporated into the production model as exogenous (uncontrollable) inputs or outputs, or as categorical variables (*c.f.* [ALI93]); and may also be used at a later stage for hypothesis testing.

It is important to note that Reese and Paradi *et al.* utilized input-oriented DEA models. In most situations the main focus of the analysis of the software production processes is on the cost savings associated with efficiency improvements. Furthermore, the outputs of the software project, such as functionality, are usually determined by the market, clients or users. Thus, input-oriented models are typically the most appropriate for evaluating software production processes.

3.3. An Enhanced Model of Software Production

While the model presented in [REES93] and [PARA95] has proven to be useful to IS management in several instances, it is limited to evaluating the performance of "similar" software projects. Similar projects would be those, for instance, that produced entirely new code. This model is not as well suited for analyzing maintenance projects since the degree of unmodified code and total system size is not taken into account. Furthermore, the project size may be measured as an aggregate of both new and modified code.

This aggregation limits our ability to properly select meaningful efficient peers when benchmarking performance using DEA, since it does not allow for the proper discrimination between new development and modification activities. When modifying code, the programmer is left with the legacy of the previous design and software, and must work with this system to adapt, or enhance it by adding new code or modifying existing code, etc. For this and many other reasons, modifying and enhancing software can be much more difficult and expensive, for the same amount of code, than developing new code (*c.f.* [JONE91]).

In situations where projects involve varying degrees of maintenance or enhancement, it is also necessary to consider the total size of the system being maintained when establishing efficient resource benchmarks for the observed project outcomes. For instance, a project with 200 new and modified FP with a total system size of 500 FP is quite a different project than a project with 200 new and modified FP and a total system size of 2000 FP. The main difference is in the greatly increased effort required to conduct the requirements analysis and design of the additional code in the context of the much larger total system. Furthermore, adding or modifying 200 FP of a much larger 2000 FP and then testing the *entire* code would likely require much more total effort.

The main reason for the increased effort is the larger size and complexity of the 2000 FP system. Moreover, although the added and modified code is only 200FP (what amounts to 10% of the total system), many requirements analysis, design and testing activities must be conducted on most of, or even, the entire system. Clearly, the total system size is an important variable that must be considered in order to properly assess the performance of projects adding or modifying existing software. This issue is further addressed in Section 3.3.1.

This enhanced production model is as follows:

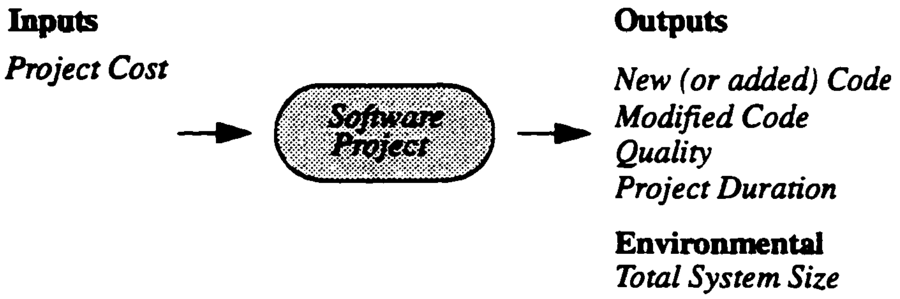


Figure 3-2 An Enhanced Model of Software Production

The inputs and outputs listed in Figure 3-2, can each be measured in several ways as discussed in the previous section. For instance, *new code*, *modified code* and *total system size* can be measured using Function Points or Source Lines of Code. Note that the quality and duration measures may have to be transformed as also discussed in the previous section.

The main difference between this model and the previous software production model of Reese and Paradi *et al.* is the inclusion of an environmental variable, *total system size*, which ensures that projects are compared to efficient peers that are of a similar total size. Furthermore, project size is now captured in two parts as *new* and *modified* code. These changes result in much more meaningful comparisons for those projects with substantial unmodified portions of code. This also means that development projects and projects involving varying degrees of maintenance can be analyzed (or benchmarked) together in an appropriate manner.

3.3.1. Numerical Illustration

The following numerical example illustrates the importance of incorporating the total system size variable in the enhanced production model of Figure 3-2. It is shown how this can have a favourable impact on the selection of efficient peers and on the DEA efficiency scores as well.

The data set used in this example, listed in Table 3-2, was drawn from a large Canadian bank and contains thirteen completed projects. Four of the projects represent new development, while the remaining nine represent enhancement type maintenance (not simply bug fixes). This maintenance constitutes varying degrees of enhancements (new code added) as well as substantial portions of existing code that has been modified. Notably, six of the nine maintenance projects have sizeable portions of unmodified code. Unmodified code is found by subtracting the *new* and *modified* code totals from the *total system size*. The duration measure was transformed as discussed in Section 3.2. for analysis purposes. Table 3-2 lists the raw, untransformed data.

Due to the small sample size the *new* and *modified* code variables have been aggregated and the *quality* measure has not been included due to incomplete and inconsistent data. Ideally, with a larger sample size, the *new* and *modified* outputs would not have had to be aggregated, thus allowing for better discrimination between development and maintenance activities. However, this data set will still serve to illustrate the importance of including the total system size variable.

Table 3-2 Efficiency Scores for Different Models

Project Number	Project Data				Efficiency Score Comparison	
	Effort (Work Months)	New & Modified Size (FP)	Duration (Months)	Total System Size (FP)	Basic model of Paradi et al.	Enhanced (VRS)
1	96	531	21	805	0.37	0.42
2	31	78	7	78	1	1
3	55	227	5	227	1	1
4	53.5	271	23	271	0.55	0.55
5	27.25	168	16	168	1	1
6	70.5	2032	13	2032	1	1
7	55	282	12	1660	0.59	1
8	189	1639	14	5287	0.32	1
9	129.5	315	13	315	0.25	0.25
10	325.25	1716	24	1716	0.19	0.19
11	73	75	11	250	0.40	0.43
12	70	169	36	228	0.39	0.41
13	124	162	28	677	0.22	0.30

The cost of each project is approximated by the total effort (labour). The total project cost is normally calculated at the Bank by multiplying the total effort by a fixed chargeout rate and adding any extraordinary items such as consultant fees etc. Only the total effort for each project was available from the cooperating Bank.

The basic model of Reese depicted in Figure 3-1 was used to first evaluate the data set of thirteen software projects. The total system size variable is not included in the model specification and a quality output was also not included, as mentioned above. The results of a DEA analysis using this model and an input-oriented VRS model (BCC-I) are given in Table 3-2 in the second last column. Notice that projects 2, 3, 5 and 6 were found efficient - projects with no unmodified code. In fact, all of the projects with unmodified code (projects 1, 7, 8, 11-13) were found highly inefficient, all with efficiency scores under 0.60.

The results for the enhanced production model are listed in Table 3-2 in the last column. This model incorporates the total system size as an environmental variable and was also evaluated with the a BCC-I DEA model. As expected, the inclusion of the *Total System Size* variable only modified the efficiency scores of the six projects that had a portion of the total system that was unmodified (projects 1, 7, 8, 11-13). The efficiency score of each of these six projects increased on average by 21%, reflecting the modification in peer group (see APPENDIX A for complete results). Furthermore, projects 7 and 8 were now rated as efficient, in sharp contrast to the results using the Basic Production model. By including this environmental variable, we allow for the extra effort required to add and modify only a portion of a larger system. If this were not done, then one would be assuming that adding or modifying a portion of a larger system would require the same amount of effort as developing or modifying the same size complete system. Further discrimination can be provided between similar size development and maintenance efforts if a larger data set were available and new FP and modified FP were not aggregated. These results demonstrate the importance of incorporating the *Total System Size* variable into the performance evaluation.

3.4. A Multi-Stage View of Software Production

The multi-stage model presented by Banker et al. [BANK91B] divides the software maintenance process into two phases (see page 39). The advantage of this multi-stage view of software production is that it supports analysis of the effects of project characteristics on each phase of the software project life cycle. However, there exist other, more general (multi-stage) views of software production.

Pressman [PRES92], for instance, views all *software engineering processes* generically. Pressman defines software engineering processes as an integration of *methods*, *tools*, and *procedures* for the development of computer software. A software engineering *paradigm* is a set of steps that encompasses the software engineering methods, tools and procedures. Wortman [WORT94] lists some of the current software engineering paradigms:

- the Wild West Approach;
- Waterfall Model (the Classic Life Cycle Model);
- Rapid Prototyping;
- Spiral Model;
- Software Reuse.

Pressman argues that *any* software development process, regardless of paradigm, application area, size or complexity, contains the following three phases:

- definition (*i.e.* analysis);
- development;
- maintenance.

Here, definition includes system and requirements analysis, as well as software project planning. Development contains three specific steps: *design*, *coding*, and *testing*. Software maintenance reapplies the steps of definition and development, but does so in the context of existing software systems.

Thus, software development, maintenance, and even the complete software life cycle, can be viewed as analysis and "development" activities. Moreover, at a higher level of abstraction, these processes could also be viewed as analysis, design, coding and testing activities. Following Banker et al. [BANK91B], the analysis and design phases could be combined in a production model since quantifying the output of analysis is quite difficult. Furthermore, analysis and design can be difficult to isolate from each other in real software projects.

Building on the work of Reese and Paradi *et al.*, and following Pressman, a three stage software production model can be constructed which also incorporates the critical attributes of *cost*, *size*, *quality* and *duration* as well as *total system size*. This model, depicted in Figure 3-3 as an SADT diagram, imposes the enhanced production model depicted in Figure 3-2 on a three stage software production process containing analysis/design, coding and testing phases. For each phase, as in Reese and Paradi *et al.*, total cost includes the cost of all activities (including overhead etc.), size of the design and software is measured in FP, quality is measured as either a defect count or rework hours, and duration is measured as the calendar duration of each phase. The result is three separate but related production models, one for each phase, that have a single input (cost), four discretionary outputs (new code size, modified code size, quality, duration) and one non-discretionary output (total system size).

Note that the quality measures for a particular phase will not be available until the following phases, or time period. For instance analysis errors (*e.g.* misspecification) and design errors will be found in the coding and testing phases. Coding errors will be found in the testing phase. Finally, defects that escape the testing phase will be found when the system is on-line and being used. Note that the definitions of quality applied with this model are very narrow and can be greatly improved (see Section 3.2.). The issue of the cost of the analysis/design phase must also be clarified. Ideally, software engineering starts after systems engineer-

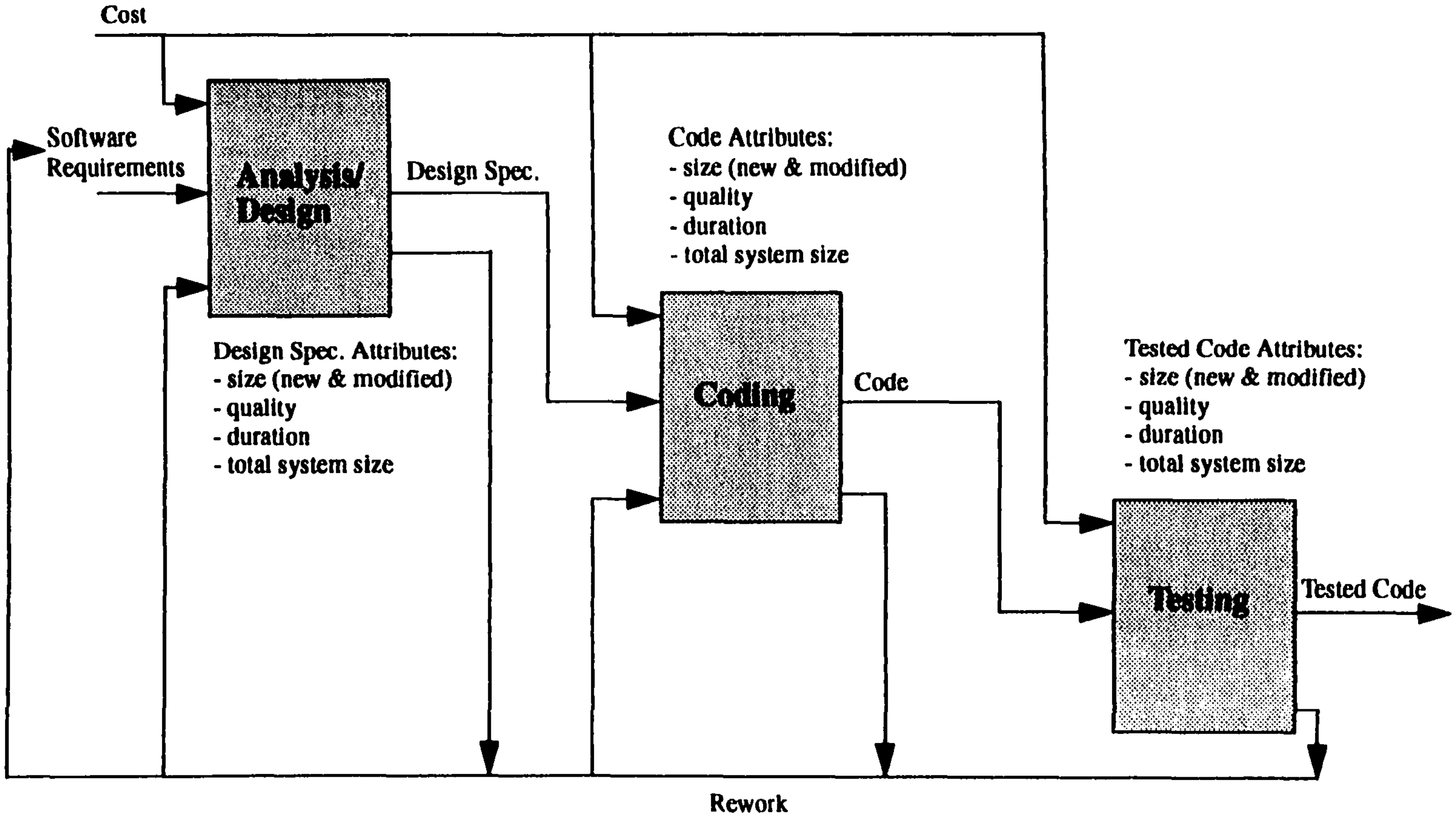


Figure 3-3 Three Stage Model of Software Production (SADT Diagram).

ing: the process that includes defining the scope of the information system and allocating functionality between hardware, software and people. At this point, software specification begins and then so too should costing for the software analysis/design phase. However, there will exist cases when this separation between systems and software costing will not be possible. In this case analysis/design costs, will include systems engineering costs. Consequently, one must be careful to define the boundary of the production model, and, hence, the boundary of the analysis, in a consistent manner.

A multi-stage software production model can be advantageous when compared to single stage production models. This stems from the following assertion: different methods and technologies are used for each of the analysis/design, coding and testing phases (*c.f.* [**PRES92**]). Many different approaches can be used for analysis and design such as structured analysis and data flow oriented design, object-oriented analysis and design, and Jackson System Development. These techniques can be independent of the coding methods and technologies used, such as procedural or object oriented programming languages, code reuse techniques and libraries, etc. Finally, testing is conducted usually by a team independent of the software developers. Testing also requires special expertise and specific techniques. Furthermore, software testers can utilize many types of support tools including test file generators, profiling tools and on-line 'debuggers'.

A multi-stage model can also be used to complement a single stage production model in order to evaluate best practices and technologies that are *phase-specific* (specific to the coding phase, for instance). To do this, it is critical that management be able to "drill down" to the sub-process or phase level. Moreover, a multi-stage model may provide insights on how to improve software teams or projects that are found to be efficient under single stage models. Additional insights may also be provided for teams or projects found inefficient using single stage production models as well.

The main strength of this multi-stage model is that the software subprocesses, or phases, can be viewed in a generic manner. However, it must be pointed out that in this multi-stage model, each subprocess model is prone to the same strengths and weaknesses as the previous single-stage model (Section 3.2.).

No data was available to illustrate the application of the multi-stage production model at this time. There are not many software producing organizations that have sophisticated measurement programs currently in place [GIBB94]. Furthermore, not all producers measure software size with FPs. In fact, not many firms that do measure size with FPs capture the size of the system specified in the software design (this is useful for early cost estimation). Thus, this model has limited use at present in many software producing organizations.

To address these issues, the three stage model can be modified in order that it is more relevant to the current situation of most producers of software. The analysis/ design and coding phases can be combined to simplify the model and its data requirements. In doing so, this eliminates the need to measure the size of the design specification and also allows for the size of the completed and tested code to be assessed with other measures such as SLOC - still a common measure. Moreover, this approach is applicable to many of the current methodologies and approaches used to develop and maintain software where it is difficult to isolate the analysis/design and coding phases. Organizations, such as the large Canadian Banks, that utilize a different team or part of the organization to perform final independent (including integration) testing would find the two stage model immediately applicable and immensely useful. The performance of the testing processes could be evaluated in a manner consistent with the development and maintenance processes. It is not uncommon for up to 60% of software development costs to be allocated to software testing in the large Canadian Banks.

3.5. Summary

In this chapter, models of software production presented in the literature are first reviewed. Of particular interest is the production model of Reese [REES93] and Paradi, Reese and Rosen [PARA95] which considers the project attributes of total cost, size, quality and duration. Also of significance is the two-stage model of software maintenance presented by Banker *et al.* [BANK91B]. It is also argued that input-oriented DEA models are typically the most appropriate for assessing software production processes since the outputs of the process are usually determined by the market, the user or the client.

An enhanced production model, based on this previous work, is presented that includes the environmental or nondiscriminatory variable *Total System Size*. By including this environmental variable, we allow for the extra effort required to add and modify only a portion of a larger system. If this were not done, then one would have to assume that adding or modifying a portion of a larger system, would require the same amount of effort as developing or modifying a complete system of the same size.

Further discrimination is provided between similar size development and maintenance efforts by including both *new* code and *modified* code as separate output variables, as opposed to the earlier model of Reese and Paradi *et al.* where these two variables are aggregated. A numerical illustration demonstrates the importance of incorporating the *Total System Size* variable into the performance evaluation process. This view of software projects represents a much more sophisticated view of software processes than the simple classification of projects as development or maintenance for measurement and analysis purposes. Furthermore, incorporating information regarding unmodified code (in the total system size variable) also makes the model appropriate for many situations where code is re-used, purchased, drawn from object libraries and so forth.

Pressman's [PRES92] definitions of software engineering processes and paradigms provide a useful characterization and an excellent starting point for developing multistage software production models. Pressman argues that all software development processes can be viewed generically: regardless of paradigm, application area, size or complexity since all development processes contain definition (requirements analysis), development (design and coding) and testing phases.

It has been shown how this view of software, along with the new enhanced production model presented herein can be combined into a generic three stage view of software production. This new multi-stage model of software production contains three phases: *analysis/design*, *coding*, and *testing*. The main advantage of this model is that it can be used to evaluate best-practices and technologies at the phase or sub-process level. This sub-process view is a critical one since different methods and technologies are employed at each phase. Furthermore, this multi-stage view of software production can be used in conjunction with the single-stage model to investigate phase-specific technologies and performance issues.

For some organizations the three stage model may require too rigorous and too large a measurement program, or may not be applicable given the methods and processes used to produce software. Thus, a simpler two stage model has also been presented which contains an analysis/design/coding phase and a testing phase. This allows for the separate evaluation of the software producing and testing processes which often utilize very different and specialized technology, usually applied by different teams in the organization. With the high costs of software testing, and the high priority given to software quality, this model may prove immediately applicable and useful to many organizations.

In some software producing organizations, software is produced to different levels of engineering rigor. For instance, a management information system presenting quarterly financial reports does not need to be designed and built to the same engineering standards as a safety and control system for a nuclear reactor.

Future work will attempt to incorporate levels of engineering rigor into the software production models.

Finally, as software producing organizations mature, the software process capability will likely increase. Accordingly, the needs of the performance measurement program will change to reflect the process improvements. Future work will also look at presenting software production models for different levels of process maturity, such as the five levels defined in the Software Engineering Institute's Capability Maturity Model (see [HUMP89] and [PAUL95]).

CHAPTER 4 *Measuring Overall Efficiency and Effectiveness Using DEA*

"There is nothing so useless as doing efficiently that which should not be done at all." *Peter F. Drucker*

4.1. Introduction

Performance measurement and assessment is fundamental to management planning and control activities, and accordingly, has received considerable attention by both management practitioners and theorists. Data Envelopment Analysis (DEA) is now well established as a theoretically sound framework for conducting performance analysis, and its application by practitioners has resulted in some significant performance improvements (*c.f.* [SHER95]). Furthermore, these methods possess many advantages over traditional techniques such as performance ratio and regression analysis. This makes DEA a very suitable tool for software engineering management.

The basic DEA models do not require *a priori* specification of input and output weights (also referred to as multipliers) and by letting these weights run freely, estimates of *technical efficiency* are obtained. However, in practice, it may be desirable to place restrictions on the weights for the following reasons:

- the multipliers, if left to run freely, may take on some values that are unreasonable, leading to overly optimistic, even unrealistic, measures of (technical) efficiency;

- to measure the *effectiveness* of achieving one or more behavioural objectives, multipliers must reflect realistic values or prices.

In management literature, efficiency is often associated with performing activities as well as possible, whereas effectiveness is often equated with the proper selection of the activities, or with doing the "right things" (*c.f.* [DRUC77], [GRIF87], [ANTH89]). Thus, an organization, business unit or DMU, is effective to the degree to which it achieves its goals. Measures of effectiveness evaluate the performance of business unit efforts with respect to strategic goals, and serve as a critical component in management planning and control processes [GRIF87]. While several different approaches have been developed to model organizational effectiveness (*c.f.* [HALL91]), there appears to be a lack of analytical tools to measure and assess effectiveness.

Several DEA research efforts have explicitly addressed the critical management issue of effectiveness analysis (*c.f.* [GOLA88], [KORN91], [GOLA93]). For instance, Golany proposes that effectiveness measures how close a DMU performs to some given goal(s) or objective(s) and argues that inefficiency is associated with waste and, therefore, cannot be associated with effective operations [GOLA88]. Moreover, many other DEA techniques exist that can be applied to assess DMU effectiveness with respect to given goals and objectives.

When precise (monetary) prices exist and are known, DEA models can be used to gauge *overall efficiency*. These models measure the degree to which a single behavioural objective such as cost minimization or revenue maximization has been attained [FÄRE94]. Clearly, overall efficiency (as introduced in Section 2.4.) can be seen as a special case of effectiveness. When the behavioural objective is profit maximization, both DEA models (*c.f.* [LOVE93]) and other non-parametric methods, the so called "Dual Approach" (see [BANK88] and [CHAV94]), can be applied to measure overall profit efficiency. These models are strongly linked to traditional economic production analysis where deviations from overall efficiency are disaggregated into technical and allocative efficiency measures (see [FARR57]).

In the absence of precise prices or other value measures, models incorporating weight constraints such as cone-ratio (CR) DEA models (see **[CHAR90]**) can be used to assess effectiveness. Nonetheless, it is unclear which CR model is appropriate for a particular behavioural goal. Weight constraints may be based on more objective information such as price ranges (*c.f.* **[THOM94]**) or more subjective information such as individual or group judgement, or measured preference for multiple organization goals (*c.f.* **[PARA95]**).

However, what is lacking in the literature is a clear explanation as to the exact relationship between these approaches; the CR models and the economic production analysis approach. Furthermore, it is unclear what exactly is being measured from the perspective of economic production analysis when multipliers are restricted (*i.e.* bounded) to reflect behavioural goals. A desirable property would be that as the multiplier cones get more restrictive, the resulting objective function converges to a well defined measure of overall efficiency.

This chapter presents a framework for the application of DEA to measure overall efficiency. Furthermore, it is shown how this framework can be applied to assess effectiveness for more general behavioural goals: those beyond the scope of overall efficiency. First, the relationships between various cone-ratio models and models to measure overall efficiency are clarified. Specifically, it is shown that as multiplier cones (based on market prices) tighten, the cone-ratio models converge to measure overall efficiency. Conditions for equivalence are also established with the so-called Dual Non-Parametric Methods under a single fixed set of market prices. Furthermore, it is argued that multiplier cones and cone-ratio model selection must be consistent with the behavioural goals assigned or assumed for purposes of analysis. Consistent with this reasoning, two new models are introduced to measure effectiveness when value measures are represented by separable or linked cones. The latter is particularly useful for analyzing profit-maximizing effectiveness, as well as measuring effectiveness when goal representation requires linked cones.

Since most organizational goals cannot be represented solely by market prices, measuring effectiveness often requires that subjective information regarding managerial goals be added to the model. Caution must also be taken to ensure appropriate DEA model selection. Thus, as stated earlier, it is shown how this framework can be applied to assess effectiveness for these more general behavioural goals.

4.2. The Relationship Between Weight Flexibility and Overall Efficiency

In this section the relationship between the input and output-oriented cone ratio models and overall cost and revenue efficiency is formally established. It is also shown that input and output-oriented linked cone models are equivalent to the reciprocals of input and output distance functions defined by (EQ 2-30) and (EQ 2-31) respectively. Furthermore, a new linked-cone model (and corresponding efficiency measure) is developed that converges to measure overall profit efficiency (as in (EQ 2-28)). Several major problems are identified with this model and its measure of profit efficiency, thus, making them both impractical for use in many situations.

In order to investigate the nature of this relationship, it is useful to introduce the following notation and terminology. Recall that a cone W can be represented in sum form, as a linear combination of extreme vectors $\mathbf{b}_i$. This cone can be equivalently constructed using the rectilinear norms of $\mathbf{b}_i$.

$$\text{Let } \bar{\mathbf{b}}_i = \frac{\mathbf{b}_i}{\sum_i \mathbf{b}_i}, \forall i \text{ and } \bar{\mathbf{B}}^T = (\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_l) .$$

where, $W = \{\mathbf{B}^T\alpha : \alpha \geq \mathbf{0}\} = \{\bar{\mathbf{B}}^T\alpha' : \alpha' \geq \mathbf{0}\}$ by definition.

The distance between any two vectors in W can be measured using a standard distance function: $d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(x_{11}-x_{21})^2 + (x_{12}-x_{22})^2 + \ldots + (x_{1m}-x_{2m})^2}$. Furthermore, one can characterize the restrictiveness of a cone by the distances between the normed extreme vectors (or generators).

Definition 4.1: A multiplier cone $W = \mathbf{B}^T\alpha, \alpha \geq \mathbf{0}$ with $Int(W) \neq 0$ *tightens* when $\mathbf{B}^T\alpha \rightarrow \mathbf{p}\alpha$, for any $\mathbf{p}\alpha \subset W, \alpha \geq 0$, as $d(\bar{\mathbf{b}}_i, \bar{\mathbf{p}}) \rightarrow 0, \forall i$, where $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$, and $\bar{\mathbf{p}} = \mathbf{p}/[\sum_i \mathbf{b}_i]$.

This definition can be applied to relate several cone-ratio models to specific measures of overall efficiency. We will consider only VRS models in this and following sections due to the favourable feasibility characteristics of these models. Refer to APPENDIX B for a detailed discussion of feasibility issues.

Theorem 4.1: As the input (price) cone V of an input-oriented cone-ratio model *tightens*, the objective function converges to the measure of overall cost efficiency.

Proof: Let us begin by considering a sum form input-oriented cone-ratio model, with the tightest possible input cone $\mathbf{c}\alpha$, $\alpha \geq 0$, for a given input price vector $\mathbf{c}$:

$$\begin{aligned} \underset{\theta,\lambda}{\text{Min}} \quad & \theta \\ \text{s.t.} \quad & \mathbf{Y}\lambda \geq \mathbf{y}_0 \\ & \theta\,[\mathbf{c}^T\mathbf{x}_0] \geq \mathbf{c}^T\mathbf{X}\lambda \\ & \mathbf{1}^T\lambda = 1 \\ & \lambda \geq \mathbf{0} \end{aligned} \qquad \text{(EQ 4-1)}$$

If the following substitution is made into (EQ 4-1):

$$\theta = \mathbf{c}^T\mathbf{x}/\mathbf{c}^T\mathbf{x}_0$$

then we obtain the following linear program:

$$\min_{x,\lambda} \frac{\mathbf{c}^T\mathbf{x}}{\mathbf{c}^T\mathbf{x}_0} \tag{EQ 4-2}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{Y\lambda} \geq \mathbf{y}_0 \\ & \mathbf{c}^T\mathbf{x} \geq \mathbf{c}^T\mathbf{X\lambda} \\ & \mathbf{1}^T\mathbf{\lambda} = 1 \\ & \mathbf{\lambda} \geq \mathbf{0} \end{aligned}$$

Contrast this to a basic model to measure overall cost efficiency (OE_i) for DMU_0:

$$\min_{x,\lambda} \frac{\mathbf{c}^T\mathbf{x}}{\mathbf{c}^T\mathbf{x}_0} \tag{EQ 4-3}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{Y\lambda} \geq \mathbf{y}_0 \\ & \mathbf{x} \geq \mathbf{X\lambda} \\ & \mathbf{1}^T\mathbf{\lambda} = 1 \\ & \mathbf{\lambda} \geq \mathbf{0}. \end{aligned}$$

The solution spaces of the programs of (EQ 4-2) and (EQ 4-3) differ in dimensionality by (m-1), and, thus, the programs are not equivalent. However, to complete the proof, it is sufficient to show that the objective function spaces are equivalent. Accordingly, the two programs will be represented in set notation. Let:

$$L(\mathbf{y}) = \{\mathbf{x} : (\mathbf{y}, \mathbf{x}) \in T\}, \tag{EQ 4-4}$$

$$L'(\mathbf{c}, \mathbf{y}) = \{\mathbf{c}^T\mathbf{x} : (\mathbf{y}, \mathbf{x}) \in T\},$$

where $L(\mathbf{y})$ represents the production technology as an input set and $L'(\mathbf{c}, \mathbf{y})$ is the set of feasible costs associated with producing output vector **y** for a given cost vector **c**. The program given by (EQ 4-3) can be represented as:

$$OE_i = min\{L'(\mathbf{c},\mathbf{y})\}/(\mathbf{c}^T\mathbf{x}_0). \tag{EQ 4-5}$$

Furthermore, the program given by (EQ 4-2) can be represented as:

$$min\{\mathbf{c}^T\mathbf{x} : \mathbf{c}^T\mathbf{x} \in L'(\mathbf{c},\mathbf{y})\}/\mathbf{c}^T\mathbf{x}_0. \tag{EQ 4-6}$$

Clearly, (EQ 4-6) can be simplified to the form of (EQ 4-5). Thus, the programs represented by (EQ 4-3) and (EQ 4-2) have the same optimal objective function, provided that a feasible solution exists for both. □

Theorem 4.2: As the output (price) cone U of an output-oriented cone-ratio model *tightens*, the objective function converges to the measure of overall revenue efficiency.

Proof: Similar to that of Theorem 4.1 □

Let us now consider a model to measure overall profit efficiency, based on (EQ 2-28), where throughput vector $\mathbf{z}^T = (\mathbf{y}^T, -\mathbf{x}^T)$, and $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$:

$$\begin{aligned} &\underset{z,\lambda}{\text{Max}} \ \frac{\mathbf{p}^T\mathbf{z}}{\mathbf{p}^T\mathbf{z}_0} \\ &\text{s.t.} \quad \mathbf{Z}\boldsymbol{\lambda} \geq \mathbf{z} \\ &\qquad\ \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ &\qquad\ \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \tag{EQ 4-7}$$

If the solution space is transformed using the revenue and cost vector $\mathbf{p}$ in the following manner:

$$\begin{aligned} &\underset{z,\lambda}{\text{Max}} \ \frac{\mathbf{p}^T\mathbf{z}}{\mathbf{p}^T\mathbf{z}_0} \\ &\text{s.t.} \quad \mathbf{p}^T\mathbf{z} \leq \mathbf{p}^T\mathbf{Z}\boldsymbol{\lambda} \\ &\qquad\ \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ &\qquad\ \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \tag{EQ 4-8}$$

it can be readily shown that the resulting linear program, while not equivalent to (EQ 4-7), has the same optimal objective function. If the following substitution is then made into (EQ 4-8):

$$\pi = \frac{\mathbf{p}^T\mathbf{z}}{\mathbf{p}^T\mathbf{z}_0}$$

then we obtain the following linear program:

$$\begin{aligned} \underset{\pi,\lambda}{\text{Max}} \quad & \pi \\ \text{s.t.} \quad & \pi[\mathbf{p}^T\mathbf{z}_0] \le \mathbf{p}^T\mathbf{Z}\boldsymbol{\lambda} \\ & \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ & \boldsymbol{\lambda} \ge \mathbf{0} \end{aligned} \tag{EQ 4-9}$$

The program of (EQ 4-9) constitutes a new (cone-restricted) radial model which measures efficiency via throughput (or netput) augmentation. Furthermore, this model is consistent with the notion of overall profit efficiency, as defined in (EQ 2-28), since the efficiency measures of this model are the same as those of (EQ 4-7). The program of (EQ 4-9) can be generalized to incorporate any sum-form cones in the following manner:

$$\begin{aligned} \underset{\pi,\lambda}{\text{Max}} \quad & \pi \\ \text{s.t.} \quad & \pi[\mathbf{B}\mathbf{z}_0] \le \mathbf{B}\mathbf{Z}\boldsymbol{\lambda} \\ & \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ & \boldsymbol{\lambda} \ge \mathbf{0} \end{aligned} \tag{EQ 4-10}$$

This derivation clearly establishes the role of linked-cones in measuring overall profit efficiency, since the relative value of both inputs and outputs must reflect market prices.

However, in measuring overall profit efficiency (as defined in (EQ 2-28)), (EQ 4-9) also possesses similar problems and limitations. Moreover, this new model measure also provides some further insights into inherent problems and difficulties. For instance, it is clear that the method of projection (to the frontier) augments both the inputs and outputs proportionately (this projection is also inherent in (EQ 2-28)). While this projection will result in an increase in profit through simple scaling, it is more desirable to increase outputs (or revenues) and decrease inputs (or costs). Furthermore, the weight constraints appended to (EQ 4-9) may result in an unbounded linear program since the projection may be in a direction away from the (constrained) frontier. The dual model given in APPENDIX B also reveals an inherent restriction of $\omega \geq 1$. This is related to the built-in restriction of positive profit in the overall profit efficiency definition of (EQ 2-28).

Now consider a sum form linked cone model similar to (EQ 4-9) but with an input-oriented efficiency measure and the tightest possible linked-cone **p**:

$$\begin{aligned} &\underset{\theta,\lambda}{\text{Min}} \quad \theta && \text{(EQ 4-11)} \\ &\text{s.t.} \quad \mathbf{p}^T \begin{bmatrix} \mathbf{y}_0 \\ -\theta\mathbf{x}_0 \end{bmatrix} \leq \mathbf{p}^T \begin{bmatrix} \mathbf{Y} \\ -\mathbf{X} \end{bmatrix} \boldsymbol{\lambda} \\ &\qquad \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ &\qquad \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Contrast this to the following model used to calculate the input distance function defined by (EQ 2-30), based on the non-parametric frontier L defined by (EQ 2-26):

$$\text{Min } h \qquad \text{(EQ 4-12)}$$

$$\text{s.t.}\quad \mathbf{p}_j^T\begin{bmatrix}\mathbf{y}_0\\-h\mathbf{x}_0\end{bmatrix} \le \mathbf{p}_j^T\begin{bmatrix}\mathbf{y}_j\\-\mathbf{x}_j\end{bmatrix},\ \forall j \in E.$$

where E is the set of DMUs which maximize profit (see Section 2.4.).

Theorem 4.3: As the linked cone of an input-oriented linked cone ratio model tightens (as in (EQ 4-11)), the linear program converges to equivalence with that of (EQ 4-12), the reciprocal of the non-parametric input distance function of (EQ 2-30) with a single (fixed) set of prices.

Proof: Let $num(E)$ represent the number of elements in a given set E. Consider (EQ 4-12) under the condition of one fixed set of prices such that $\mathbf{p}_j = \mathbf{p}, \forall j \in E$. Two cases must be considered: if $num(E) > 1$ and if $num(E) = 1$. If $num(E) > 1$ then:

$$\mathbf{p}^T\begin{bmatrix}\mathbf{y}_i\\-\mathbf{x}_i\end{bmatrix} = \mathbf{p}^T\begin{bmatrix}\mathbf{y}_j\\-\mathbf{x}_j\end{bmatrix},\quad i, j \in E, i \ne j,$$

and (EQ 4-12) contains redundant constraints. Thus, the linear program in (EQ 4-12) when $num(E) > 1$ is equivalent to the following linear program where the production possibilities are defined by a single efficient DMU_j, where $j \in E$:

$$\text{Min } h \qquad \text{(EQ 4-13)}$$

$$\text{s.t.}\quad \mathbf{p}^T\begin{bmatrix}\mathbf{y}_0\\-h\mathbf{x}_0\end{bmatrix} \le \mathbf{p}^T\begin{bmatrix}\mathbf{y}_j\\-\mathbf{x}_j\end{bmatrix},\ \text{for a single } j \in E.$$

Now consider (EQ 4-11) and let:

$$E' = \left\{ j | \theta_j = 1, \mathbf{p}^T \begin{bmatrix} \mathbf{y}_j \\ -\mathbf{x}_j \end{bmatrix} = \mathbf{p}^T \begin{bmatrix} \mathbf{Y} \\ -\mathbf{X} \end{bmatrix} \lambda_j \right\}$$

where the subscript *j* refers to the optimal values of decision variables solved for a particular DMU_j. Set E' contains those DMUs which maximize profit and, thus, $E' = E$ by definition. If $E' > 1$ then the production possibility set is defined by convex combinations of the same maximum observed profit. Clearly, (EQ 4-11) can simplified to the form of (EQ 4-13).

A similar proof can be constructed for the case where $num(E) = 1$. □

Theorem 4.4: As the linked cone of an output-oriented linked cone ratio model tightens, the linear program converges to equivalence with the reciprocal of the output distance function defined by (EQ 2-31) with a single (fixed) set of prices.

Proof: Similar to that of Theorem 4.3. □

These two proofs imply that a sum form cone-restricted production possibility set is identical to the "upper bound" production possibility set under a fixed set of prices. This further strengthens the link between DEA the traditional non-parametric production analysis literatures (reviewed in Section 2.4.).

4.3. Some New Models and Efficiency Measures

Recall that it is possible to classify cones into four different types: ***input***, ***output***, ***separable*** and ***linked***. Thus far, the relationship between various input, output and linked-cone DEA models and well defined measures of overall efficiency have been examined and clarified. However, many problems exist with the previous linked-cone model (EQ 4-9) and (EQ 4-10), as well as its inherent definition and measure of profit efficiency, making it impractical for use in many situations (see Section 2.4. and Section 4.2.). Thus, new measures and more practical linked-cone DEA models must be developed for application to profitability analysis. Furthermore, one last type of cone has yet to be addressed and its role in measuring overall efficiency clarified - the *separable* cone. Specifically, the implicit behavioural goals and efficiency measures consistent with separable cones must be elucidated.

As stated earlier, the preferred means to improve profitability is to increase revenue and decrease costs. Thus, the associated DEA projection or efficiency measure would be one that proportionately *decreases* inputs and proportionately *increases* outputs. Furthermore, incorporation of appropriate input and output prices is necessary to ensure that performance is related to profit efficiency. For instance, if such a projection resulted in a 10% proportionate increase of outputs, then inputs should be proportionately decreased by 10%. This example can be restated in the following manner using a combination of the projections given by (EQ 2-2) and (EQ 2-7):

If $\phi = 1.1$,then $\theta = 0.9 = 1 - (\phi - 1)$.

Using this relationship, and substituting $\pi = \phi$, an efficiency measure can be defined for the general case:

$$\begin{aligned} &\mathit{max} \quad \pi && \text{(EQ 4-14)} \\ &\text{s.t.} \quad ((2-\pi)\mathbf{x}_0, \pi\mathbf{y}_0) \in T, && \end{aligned}$$

where *T* can be any valid production possibility set, such as CRS or VRS, defined in Section 2.2.

In order to apply this definition to assess overall profit efficiency, multipliers must be constrained using a cone which is consistent in form with this behavioural goal. Clearly, since profitability analysis requires market prices for both inputs and outputs, a linked cone is required in order to relate input to output multipliers. Furthermore, a VRS frontier is appropriate for most cases since a CRS frontier implicitly assumes zero maximum profit, as discussed earlier. Thus, using the efficiency measure defined by (EQ 4-14) the following linear program can be used to assess overall profit efficiency:

$$\begin{aligned} \max_{\pi,\lambda} \quad & \pi \\ \text{s.t.} \quad & \mathbf{p}^T \begin{bmatrix} \pi \mathbf{y}_0 \\ (2-\pi)\,\mathbf{x}_0 \end{bmatrix} \le \mathbf{p}^T \begin{bmatrix} \mathbf{Y} \\ -\mathbf{X} \end{bmatrix} \boldsymbol{\lambda} \\ & \mathbf{1}^T \boldsymbol{\lambda} = 1 \\ & \boldsymbol{\lambda} \ge \mathbf{0}. \end{aligned} \tag{EQ 4-15}$$

Applying the same procedure as used in the previous proofs, an equivalent ratio form of (EQ 4-15) can be derived. Thus, it is possible to show that the following has the same optimal objective function as (EQ 4-15) with the same fixed price vector $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}} \quad & \frac{\mathbf{r}^T\mathbf{y} - \mathbf{c}^T\mathbf{x} + 2\mathbf{c}^T\mathbf{x}_0}{\mathbf{r}^T\mathbf{y}_0 + \mathbf{c}^T\mathbf{x}_0} \\ \text{s.t.} \quad & \mathbf{Y}\boldsymbol{\lambda} \ge \mathbf{y} \\ & \mathbf{x} \ge \mathbf{X}\boldsymbol{\lambda} \\ & \mathbf{1}^T \boldsymbol{\lambda} = 1 \\ & \boldsymbol{\lambda} \ge \mathbf{0}. \end{aligned} \tag{EQ 4-16}$$

This model measures overall profit efficiency using the standard VRS production possibility set. Technical efficiency can be measured following definition (EQ 4-14) by using (EQ 4-15) and removing the price constraints (*i.e.* replacing vector $\mathbf{p}^T$

with an identity matrix). This measure of overall efficiency can be decomposed into a measure of technical and allocative efficiency in the usual manner (*c.f.* (EQ 2-21)).

Note that this decomposition is consistent with that proposed by Farrel (see [FARR57]) since the measure of technical efficiency does not contain prices (and does not reflect behavioural goals) and the projection (defined by (EQ 4-14)) is the same for both technical and overall efficiency measures. Furthermore, this measure of overall profit efficiency can be used in situations where DMU$_j$ has an actual profit $\mathbf{r}_j^T\mathbf{y}_j - \mathbf{c}_j^T\mathbf{x}_j \le 0$ since the measure is not based on profit ratios. Finally, this method of projection will never be in a direction away from the frontier and resulting in an unbounded linear program, when feasible multiplier or weight constraints are applied. Thus, this new measure eliminates the problems identified earlier for the measures of profit efficiency defined in Section 2.4.

We will consider one final scenario where the behavioural objective is to maximize revenue *and* minimize cost for a given set of prices $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$. For this purpose, the following linear program maximizes the *difference* between revenue and cost ratios, but does not necessarily maximize overall profit:

$$Max \ \frac{\mathbf{r}^T\mathbf{y}}{\mathbf{r}^T\mathbf{y}_0} - \frac{\mathbf{c}^T\mathbf{x}}{\mathbf{c}^T\mathbf{x}_0} \qquad \text{(EQ 4-17)}$$

$$\begin{aligned} \text{s.t. } & \mathbf{Y\lambda} \ge \mathbf{y} \\ & \mathbf{x} \ge \mathbf{X\lambda} \\ & \mathbf{1}^T\mathbf{\lambda} = 1 \\ & \mathbf{\lambda} \ge \mathbf{0} \end{aligned}$$

Notice that no direct linkage exists between revenue and cost within each ratio, in contrast to the profitability model of (EQ 4-7). One can readily show that the following linear program, while not equivalent to (EQ 4-17), has the same optimal objective function:

$$\begin{aligned} &Max\ \phi-\theta \qquad\qquad (\text{EQ } 4\text{-}18)\\ &\text{s.t. } \phi\,[\mathbf{r}^T\mathbf{y}_0] \le \mathbf{r}^T\mathbf{Y}\boldsymbol{\lambda}\\ &\qquad \theta\,[\mathbf{c}^T\mathbf{x}_0] \ge \mathbf{c}^T\mathbf{X}\boldsymbol{\lambda}\\ &\qquad \mathbf{1}^T\boldsymbol{\lambda} = 1\\ &\qquad \boldsymbol{\lambda} \ge \mathbf{0} \end{aligned}$$

The objective function of this linear program bears striking similarity to (EQ 4-17). However, in this case the multipliers are constrained with a *separable* cone. It is important to recognize the subtle but important differences between the stated behavioural goal and the objective functions of (EQ 4-17) and (EQ 4-18). Note that a model very similar to (EQ 4-18) has been presented in [ALI95].

4.4. A Framework for Measuring Overall Efficiency

The previous sections established that cone ratio DEA models with fixed input and output market prices can be applied to measure overall efficiency. However, in many cases these prices are not fixed (*i.e.* have an upper and lower bound), yet it is still desirable to measure overall efficiency. The cone ratio models specified throughout Section 4.2. with the fixed set of prices $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$, can be applied to the general case by substituting $\mathbf{p}^T = \mathbf{B}$, where $\mathbf{B}\mathbf{w} \ge \mathbf{0}, \mathbf{w} \ge \mathbf{0}$ represents a general multiplier cone (see for example (EQ 4-9) and (EQ 4-10)).

In doing so, these models can be applied to situations where market prices are not fixed and can be represented by an input, output, separable or linked multiplier cone, depending on the situation. Moreover, this extends the notion of *overall efficiency* to include the more general case of imprecise prices. Note that precise prices are represented by a special form of multiplier cone (i.e. a ray), and, thus, constitute a special case of the general multiplier cone.

Building on the results of Section 4.2. and Section 4.3. it is possible to construct a framework that prescribes the application of various DEA models contingent upon the behavioural goals and the nature of the observed market prices. This framework is summarized in Table 4-1 on page 69. Clearly, the choice of model for a given behavioural goal depends on the precision of the market prices. For instance, with precise input prices (*i.e.* costs) an input-oriented cone-ratio model with an input cone restricting the input multipliers, or the models of (EQ 2-19) and (EQ 2-20) can be used. With imprecise input prices, only the former can be used.

Table 4-1 A Framework for Measuring Overall Efficiency

behavioural Goal	Imprecise Mkt. Prices	Precise Mkt. Prices
Minimize Cost	Input-Oriented Cone Ratio (CR) Model with input cone $W \in R^m$	Use (EQ 2-19) and (EQ 2-20) (or CR Model to left)
Maximize Revenue	Output -Oriented CR Model with output cone $W \in R^s$	Use (EQ 2-22) and (EQ 2-23) (or CR Model to left)
Minimize Cost & Maximize Revenue	Use (EQ 4-18) with separable cones $V \in R^m$ and $U \in R^s$	Use (EQ 4-17) or (EQ 4-18)
Maximize Profit	Use (EQ 4-15) with linked cone $W \in R^{m+s}$	Use (EQ 4-16) or (EQ 4-15)

Both CRS and VRS technology (*i.e.* frontiers) can be utilized with each model. However, with the profit maximization models, CRS is a very restrictive assumption and may not be appropriate for many applications. Furthermore, linked cone-restrictions and a CRS frontier may result in infeasibility (see APPENDIX B).

When applying cone-ratio models to measure overall efficiency, *caution must be taken to ensure that the choice of cone (e.g. input cone) and the method of projection (e.g. input orientation) are consistent with the behavioural goal assigned or assumed for analytical purposes.* This caveat does not imply that other combinations of cones and projections are not possible or useful, but only that they do not conform to a well defined measure of overall efficiency.

4.5. Other behavioural Goals

Previously, we have considered only behavioural goals based on some function of market prices. However, in many situations where market prices do not exist, it is still of interest to measure the relative effectiveness of DMU's in meeting organizational objectives. Although developed in the context of market prices, the above framework, summarized in Table 4-1 on page 69, can also be applied for this purpose. The behavioural goal against which effectiveness is gauged is usually one of maximizing value, based on a set of organizational objectives, and based on information regarding the trade-offs between these objectives. This information is typically a set of subjective value measures and may be, in some special cases, a set of precise values.

In the latter case, some form of value function can be constructed (*c.f.* [Yu85] and [KEME87]), such as a linear additive, quadratic or Cobb-Douglas function, to measure the value of production throughputs with respect to organizational objectives. Overall maximum value could be obtained in the following fashion:

$$\begin{aligned} &Max \quad V(\mathbf{x}, \mathbf{y}) \\ &\quad \text{s.t. } (\mathbf{x}, \mathbf{y}) \in T \end{aligned} \qquad \text{(EQ 4-19)}$$

and "overall efficiency" could be measured in the traditional manner:

$$OE = \frac{V(\mathbf{x}^*, \mathbf{y}^*)}{V(\mathbf{x}_0, \mathbf{y}_0)}. \qquad \text{(EQ 4-20)}$$

When $V(\mathbf{x}, \mathbf{y})$ is a linear additive value function then the models listed in Table 4-1 on page 69 are applicable. The cost, revenue functions given in (EQ 2-19) and (EQ 2-22) are simply replaced by value functions. If $V(\mathbf{x}, \mathbf{y})$ includes all inputs and outputs, the situation is analogous to measuring maximum profitability and overall profit efficiency (assuming VRS technology). The efficiency measure defined by (EQ 4-14) is most appropriate and the model given by (EQ 4-16) should be utilized in this case.

When the value measures are imprecise, the cone-restricted models listed and classified in Table 4-1 on page 69 can be directly applied to measure effectiveness. However, careful attention should be given to the type of model to be used. For instance, if the behavioural objective is to maximize value and the multiplier cones are linked, then the model represented in (EQ 4-15) should be considered. Furthermore, VRS technology would likely be the most appropriate since CRS technology implicitly assumes a maximum value (the measure of relative effectiveness) of zero, and is not well suited for some situations.

In summary, the measure defined in (EQ 4-14) is perhaps best suited to assess effectiveness. The reason is that, in many cases, managerial objectives will involve both inputs and outputs, and it will be desirable to reduce inputs and increase outputs as opposed to increasing both. Finally, the conditions for representing the preference or value information regarding the organizational objectives as a value function are very restrictive (*c.f.* [YU85]). Thus, it is more likely that a cone-ratio model be utilized - such as (EQ 4-15).

4.6. Numerical Illustration

We now illustrate some of the strengths of the newly presented model to measure overall profit efficiency as compared to those that have previously appeared in the literature. This example is limited to two-dimensions in order that the concepts can be illustrated with standard 2-D plots. We employ a subset of the example data published in [ALI93] consisting of 11 DMUs, considering only a single input and a single output.

The data set is listed in Table 4-2 in the first three columns. The fourth column of the table, the 'zero profit price ratios', deserves some further explanation. These price ratios are simply the relative value of the cost and revenue vectors that would result in a zero profit for each respective DMU. For example, consider DMU 7 with its price ratio of 2.7. This implies that for a marginal profit (or price) of (c = -\$2.7 / *unit of x*) and a marginal profit of (r = \$1 / *unit of y*), or any combination of prices of the same proportion, the profit for DMU 7 would be:

$$-\$2.7\,(10) + \$1\,(27) = \$0.$$

This is quite useful information since it relates directly to the assumptions necessary to apply the overall profit efficiency measures of Banker and Maindiratta given in (EQ 2-28), and the related models given by (EQ 4-7) and (EQ 4-9). These measures are related to profit ratios of the following form:

(Profit of DMU i) / (Maximum Observed Profit).

The maximum overall profit and the profit of the individual DMU i must both be positive for the measure to be meaningful (recall, that (EQ 4-15) and (EQ 4-16) are based on different principles and these restrictions do not apply). Using the price ratio information given, it is possible to determine for which DMUs this profit ratio measure can be applied for a given set of prices $p = (-c, r)$. Note that a positive profit will be found for all DMUs for any set of prices with a price ratio less than that listed in Table 4-2.

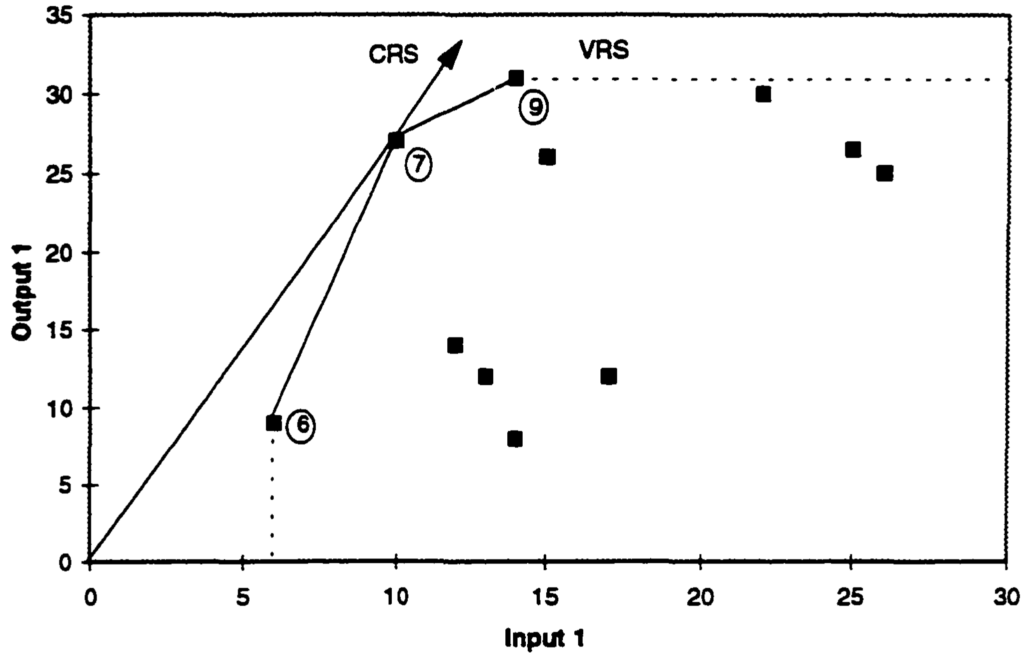
Figure 4-1 depicts the production possibility set and the corresponding VRS and CRS efficient frontiers for the example data set. Clearly, DMUs 6, 7 and 9 are VRS efficient and DMU 7 is also CRS efficient (and, thus, technically and scale efficient). Furthermore, DMU 6 exhibits increasing returns to scale (IRS), DMU 7 exhibits CRS and DMU 9 exhibits decreasing returns to scale (DRS) with a VRS frontier.

Table 4-2 Example Data Set

DMU (i)	Input 1 (x_i)	Output 1 (y_i)	Zero Profit Price Ratio $(-c/r = y_i/x_i)$
1	13	12	0.9231
2	12	14	1.1667
3	26	25	0.9615
4	15	26	1.7333
5	14	8	0.5714
6	6	9	1.5
7	10	27	2.7
8	22	30	1.3636
9	14	31	2.2143
10	25	26.5	1.06
11	17	12	0.7059

This implies, for example, that DMU 6 has a negative ω and that the virtual input $v^T x$ is larger in magnitude than the virtual output $\mu^T y$. Refer to Section 5.2. and (EQ 5-2) for the general form of the supporting hyperplanes of each facet. Therefore, the maximum overall (absolute and relative) profit of any linear profit function tangent to DMU 6 will be negative. This is the case since the profit function will have the same ω (or range) as the facet (or point) to which it is tangent. The opposite is true for DMU 9, and for DMU 7, where the maximum overall profit can either be negative, zero or positive depending on the profit function. Note that absolute and relative profit for a particular DMU differ by a positive scalar value.

Figure 4-1 Plot of Example Data Set



In the case of the CRS frontier and DMU 7, ω = 0 and, thus, $v^T x = \mu^T y$. This implies a maximum relative profit of zero if $\mu/v = r/c$. From another perspective, the only possible maximum profit function tangent to DMU 7 (under CRS) passes through the origin, parallel to the frontier, implying a zero overall maximum (absolute) profit. Both of these observations are consistent with the zero maximum profit assumption implicit with CRS frontiers (see [FÄRE88]).

We will now compare the measures of (EQ 4-7), (EQ 4-9) and (EQ 4-16) when evaluating the example data set with three different price vectors. These price vectors are selected to result in a positive, zero and negative maximum overall profit.

Table 4-3 Comparison of Different Measures of Overall Profit Efficiency Under Positive Maximum Overall Profit

DMU	Overall Efficiency for $(r, c) = (20, 10)$			Profit
(i)	(EQ 4-7)	(EQ 4-9)	(EQ 4-16)	$(ry_i - cx_i)$
1	4.3636	4.3636	2.0	$110.0
2	3.0	3.0	1.8	$160.0
3	2.0	2.0	1.3158	$240.0
4	1.2973	1.2973	1.1642	$370.0
5	24.0	24.0	2.5333	$20.0
6	4.0	4.0	2.5	$120.0
7	1.0909	1.0909	1.0625	$440.0
8	1.2632	1.2632	1.1220	$380.0
9	**1.0**	**1.0**	**1.0**	**$480.0**
10	1.7143	1.7143	1.2564	$280.0
11	6.8571	6.8571	2.0	$70.0

The first example is given in Table 4-3 and has a price vector $(r, c) = (20, 10)$. The second, third and fourth columns give the different measures of overall efficiency for this set of prices: respectively, (EQ 4-7), (EQ 4-9), which are related to a profit ratio measure, and (EQ 4-16) which is not. From column five, which lists the individual DMU profits for these prices, it is clear that DMU 9 maximizes overall profit at $480. Notice that this DMU operates under DRS and has a positive maximum profit. Furthermore, notice that none of the DMUs violates the conditions of positive profit necessary for (EQ 4-7) and, indirectly, for (EQ 4-9) as well.

As expected, the measures for (EQ 4-7) and (EQ 4-9) are identical. However, the measures differ for (EQ 4-16), except for a DMU which has an overall efficiency score of one under each approach. In this particular example, all three of the measures are useful in determining how well DMUs are maximizing profit.

Table 4-4 Comparison of Different Measures of Overall Profit Efficiency Under Zero Maximum Overall Profit

DMU (i)	Overall Efficiency for $(r, c) = (10, 27)$			Profit
	(EQ 4-7)	(EQ 4-9)	(EQ 4-16)	$(ry_i - cx_i)$
1	-inf	unbounded	1.4904	-$231.0
2	-inf	unbounded	1.3966	-$184.0
3	-inf	unbounded	1.4748	-$452.0
4	-inf	unbounded	1.2180	-$145.0
5	-inf	unbounded	1.6507	-$298.0
6	-inf	unbounded	1.2857	-$72.0
7	**undefined**	unbounded	**1.0**	**$0.0**
8	-inf	unbounded	1.3289	-$294.0
9	-inf	unbounded	1.0988	-$68.0
10	-inf	unbounded	1.4362	-$410.0
11	-inf	unbounded	1.5855	-$339.0

Table 4-4 depicts a very different situation. The prices $(r, c) = (10, 27)$ result in DMU 7 as having a maximum overall profit of zero. This profit function was selected since it is parallel to the CRS frontier. Therefore, all of the DMUs violate the conditions necessary to utilize (EQ 4-7) and (EQ 4-9). The result can be clearly seen in columns two and three. Notice that (EQ 4-16) is unaffected by the actual profit levels of the DMUs being evaluated.

Table 4-5 shows a situation similar to Table 4-4, except that all of the profits are negative. Furthermore, DMU 6 is now found to maximize overall profit and is, thus, overall efficient. Recall that DMU 6 exhibits IRS. (EQ 4-9) again was unbounded for each DMU. However, (EQ 4-7) did produce some measures in this case although the conditions for the application of the overall efficiency measures were violated. These measures can not be interpreted to those based on positive profit ratios since they span between zero and one (since the maximum profit is a smaller absolute number). Once again, (EQ 4-16) performed well, producing results that are meaningful and consistent with those of the previous examples.

Table 4-5 Comparison of Different Measures of Overall Profit Efficiency Under Negative Maximum Overall Profit

DMU	Overall Efficiency for $(r,c) = (1,5)$			Profit
(i)	(EQ 4-7)	(EQ 4-9)	(EQ 4-16)	$(ry_i - cx_i)$
1	0.3962	unbounded	1.4156	-$53.0
2	0.4565	unbounded	1.3378	-$46.0
3	0.20	unbounded	1.5419	-$105.0
4	0.4286	unbounded	1.2772	-$49.0
5	0.3387	unbounded	1.5256	-$62.0
6	1.0	unbounded	**1.0**	**-$21.0**
7	0.9130	unbounded	1.0260	-$23.0
8	0.2625	unbounded	1.4214	-$80.0
9	0.5385	unbounded	1.1782	-$39.0
10	0.2132	unbounded	1.5116	-$98.5
11	0.2877	unbounded	1.5361	-$73.0

Clearly, the overall efficiency measures (EQ 4-7) and (EQ 4-9) (and, therefore, (EQ 2-28)) also performed well under conditions where all DMUs have positive profits for a given set of market prices only; and correspond, in our example, to any region on the frontier which exhibits DRS. These measures are not useful for situations where maximum profit is zero or negative: regions which exhibit CRS and IRS. This imposes major restrictions on the applicability of these methods to many real world applications where positive profits may not exist. As demonstrated by these three examples, these limitations do not apply for (EQ 4-16) and, thus, (EQ 4-15).

4.7. Concluding Remarks

Throughout this chapter, it is stressed that model selection must be consistent with the behavioural goals assigned or assumed for the analysis. For this purpose, this work presents a prescriptive framework for analyzing and measuring overall efficiency. Furthermore, this framework can be applied to measure the effectiveness of DMUs in achieving behavioural or organizational objectives (including non-monetary objectives), relative to other DMUs. Notably, effectiveness is measured in a manner consistent with the Farrel framework whereby effectiveness can be decomposed into technical and allocative efficiency. This framework, along with several newly developed DEA models, directly addresses the need for more analytical tools that can provide rigorous measures of organizational effectiveness.

Overall efficiency measures the degree to which a single behavioural or organizational goal such as cost minimization has been attained for a given set of (precise) market prices. Clearly, overall efficiency can be seen as a special case of effectiveness, defined for a few very specific objectives. This approach, strongly tied to traditional economic production analysis, can provide powerful management insights. However, overall efficiency is typically defined for behavioural goals that involve precise market prices - a somewhat narrow and limited perspective. Furthermore, problems exist with the current measures of profit efficiency. As shown herein, the production analysis approach and related concepts can provide a solid foundation upon which to measure effectiveness when extended to consider other, more general, behavioural goals.

Models with general weight constraints, such as the cone-ratio models, can represent a much broader variety of behavioural goals other than monetary objectives. It is shown how cone-ratio models provide a means by which the economic production analysis approach can be extended to incorporate more general behavioural goals, provided that a framework is given to prescribe the appropriate model selection.

The framework presented herein is constructed by examining the various relationships between cone-ratio models and those that measure overall efficiency, and by building upon this knowledge. It is shown that as certain multiplier cones (based on market prices) tighten, the cone-ratio models converge to specific measures of overall efficiency. Conditions for equivalence are also established with the so-called Dual Non-Parametric Methods under a single fixed set of market prices. Furthermore, two new models are introduced to measure effectiveness when value measures are represented by separable or linked cones. The application of each model in the framework is contingent upon several factors: the behavioural goal assumed or assigned for purposes of analysis, and the degree of precision of the market prices. In doing so, the notion of overall efficiency is extended beyond the case of fixed (*i.e.* precise) market prices to incorporate a much broader range of non-monetary goals and objectives.

Although this work focuses on DEA models with VRS frontiers, the models and framework developed herein are equally applicable to CRS frontiers. Note, however, that in some applications, such as overall profit efficiency, a CRS frontier may impose overly stringent assumptions. Furthermore, while this chapter was motivated by software engineering management, the presentation was kept purposefully general. This was done since the framework and models described herein can be applied to any domain or organization where DEA analysis is appropriate.

Future work will focus on developing measures of "tightness" of multiplier cones. Work can also be done to apply the new models suited to measure profit efficiency to time series analysis, with decomposition of effects over time. This is particularly relevant to models with non-fixed prices. Furthermore, other combinations of cones and projections will be examined, and their role in measuring overall efficiency and effectiveness considered. Finally, a better connection to organization theory literature is needed for studying measures of organizational effectiveness.

CHAPTER 5 *Forecasting and Tradeoff Analysis Using DEA*

"When we try to pick out anything by itself, we find it hitched to everything else in the universe." - Muir's Law

5.1. Introduction

Software project forecasting and estimation is crucial to software project management. The project plan, which is developed at the beginning of the project and successively refined as the work progresses, defines the tasks to be completed by the project team, and provides a framework for project management and control. Furthermore, for each major task, the plan provides estimates of the time and resources necessary to complete each task. Thus, estimates of resource usage and software size are fundamental to this plan since poor forecasts and estimates can be the root cause of many schedule and cost overruns or project failures (*c.f.* **[HUMP89]**, **[PRES92]** and **[ABDE91]**). Chapter 3 reviews some of the cost estimation literature.

Software project management attempts to control organizational resources in the delivery of software functionality within the project objectives and constraints of cost, quality and time (duration). However, during the course of many projects, crises, scope changes and design changes as well as other problems upset the delicate balance needed to complete the project on time, on budget, and with the necessary quality. Changes must then be made to the project objectives, thus affecting the overall project plan. However, it is not always possible to change one objective without affecting one or more of the others. This phenomenon has been recognized in both the software literature (*c.f.* [PUTN92]) and the project management literature (see [KERZ95]).

Because of the inherent complexity of most software projects, it is necessary to develop a decision-making or tradeoff analysis process rather than rely on rigid rules or simplistic "rules of thumb" for the various project tradeoffs (*c.f.* [KERZ95]). In response to this need, some authors have called for more dynamic forecasting processes and tools (such as [YOUR96]). Such a process must also recognize that the project tradeoffs are contingent upon the specific circumstances of each project [KERZ95]. For instance, the actual time-cost tradeoffs for a very small software project could differ greatly from those for a much larger project due to a multitude of factors such as the differing size and organization of the project teams. Finally, such a process should be based directly or indirectly on actual project data, ideally from within the same organization. This can avoid portability problems and as well as the "tuning" necessary with many parametric models embedded in commercial software forecasting packages (*c.f.* [ABDE91] and [JEFF90]).

Parametric models produce forecasts based on average performance, as opposed to the best practice performance of outliers. An alternative to this approach is the use of frontier-based methods, such as DEA, to produce the forecasts as well as to analyze tradeoffs. Expert judgement or parametric models, such as those presented in [RAY91], can then be used to adjust the best practice forecast, if desired. This approach has the advantage of producing forecasts that are based on best practices and efficient processes, as opposed to averages which may be highly inefficient. As some authors have noted, inefficient forecasts can lead to inefficient project performance (*c.f.* [ABDE91]). Finally, DEA methods are non-parametric and do not impose many assumptions regarding parametric functional form that can not be directly tested [BANK88].

Most applications of DEA have focused on the assessment and control of past performance. However, several DEA applications have been for predictive purposes such as predicting bank failures (*c.f.* [BARR94], [SIEM92]) or for predicting future performance of DMU's which do not yet exist [GOLA93]. Little work has been done on applying DEA techniques for the purpose of forecasting and tradeoff analysis.

In the DEA and economic literature, tradeoffs between inputs and outputs are sometimes referred to as marginal rates - a special case of tradeoffs. Furthermore, marginal rates can be derived from optimal DEA multipliers. Mathematically, the marginal rates represent partial derivatives or slopes on the efficient frontier. Although ratios of optimal DEA multipliers also provide this information, problems of interpretation exist since the optimal multipliers may not be unique. The result is that multiplier values cannot be used directly to study marginal rates without further consideration. Rosen *et al.* directly address this problem and present a general framework for the computation of tradeoffs in DEA, as well as for the application of the multiplier information [ROSE95]. However, marginal rates are limited to assessing the impacts of *infinitesimal* changes of one or more variables on one or more other variables. As shown in [ROSE95], in the special case of DEA piece-wise linear frontiers, finite differences methods which utilize small, finite changes will also provide precise information on marginal rates. Analyzing the impacts of these very small, finite changes is not adequate for many situations where the impacts of much larger project changes are of interest. The latter is the main motivation for this chapter.

This chapter adapts the work of Rosen et al. for the purpose of developing methods to analyze project tradeoffs in an interactive manner. Specifically, the finite differences approach presented in [ROSE95] to calculate marginal rates (including returns to scale) is extended to allow analysts and project managers to interactively examine various scenarios and tradeoffs amongst the project objectives. It is also argued that the DEA production possibility set and efficient frontier can be used to deliver initial project forecasts, providing a starting point for conducting the tradeoff analysis.

5.2. Marginal Rates and DEA[1]

A DEA frontier is a piecewise linear envelopment surface made up of portions of supporting hyperplanes that form the facets of the hull constructed from the observed throughput (or netput) vectors $\mathbf{z}^T = (\mathbf{y}, -\mathbf{x})^T$. This frontier can be represented by the graph $\{\mathbf{z}: F(\mathbf{z}) = 0\}$, where the following partial derivative:

$$MR_{ij}(\mathbf{z}_0) = \left.\frac{\partial z_i}{\partial z_j}\right|_{\mathbf{z}_0} = -\frac{(\partial F)/(\partial z_j)}{(\partial F)/(\partial z_i)},\quad i \neq j \qquad \text{(EQ 5-1)}$$

is referred to as the marginal rate of throughput i to throughput j at the point $\mathbf{z}_0$ on the frontier. The marginal rate gives the increase in throughput i that results when throughput j is increased by one unit, and all other throughputs being constant (*i.e.* no change in all other throughputs).

Furthermore, let $\chi^{*T} = (\mu^*, \nu^*)^T$ denote the optimal DEA multipliers that describe a supporting hyperplane than contains a facet of the frontier. These optimal multipliers are the coefficients of the linear equations (supporting hyperplanes), of the following form, that contain the facet of the frontier:

[1] This section is based on material presented in [ROSE95].

$$\sum_{i=1}^{s}\mu_i^* y_i - \sum_{j=1}^{m}\nu_j^* x_j + \omega = 0 \qquad \text{(EQ 5-2)}$$

Note that $\omega = 0$ for CRS frontiers. Together, equations (EQ 5-1) and (EQ 5-2) imply that:

$$MR_{ij}(\mathbf{z}_0) = \left.\frac{\partial z_i}{\partial z_j}\right|_{\mathbf{z}_0} = -\frac{\chi_j^*}{\chi_i^*}, \quad i \neq j. \qquad \text{(EQ 5-3)}$$

However, due to the continuous piecewise linear nature of the DEA frontier, the marginal rates and the multipliers are not uniquely defined at all points on the frontier (*c.f.* [ROSE95]). For example, as in Figure 5-1, several (in this case two) hyperplanes may intersect at a given point on the frontier with the result that the optimal multipliers are not unique (at that point). Thus, the frontier is only piecewise differentiable at the "edges" of the frontier. As suggested by Rosen *et al.*, one solution to this problem is to take the partial derivatives, or marginal rates, to the right and to the left (respectively), as given below:

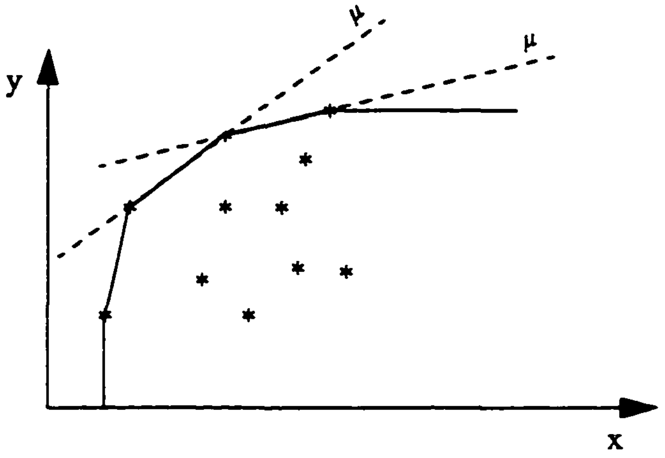
$$MR^{+}_{ij}(\mathbf{z}_0) = \left.\frac{\partial z_i}{\partial z_j}\right|_{\mathbf{z}_0^+} = \lim_{h \to 0^+}\frac{z_{i0}(z_{10}, \ldots, z_{j0}+h, \ldots, z_{(m+s)0}) - z_{i0}}{h} \qquad \text{(EQ 5-4)}$$

$$MR^{-}_{ij}(\mathbf{z}_0) = \left.\frac{\partial z_i}{\partial z_j}\right|_{\mathbf{z}_0^-} = \lim_{h \to 0^-}\frac{z_{i0}(z_{10}, \ldots, z_{j0}+h, \ldots, z_{(m+s)0}) - z_{i0}}{h}$$

where $z_{i0}(z_{10}, \ldots, z_{j0} \pm h, \ldots, z_{(m+s)0})$ is an implicit function which gives the level of z_{io} which places $\mathbf{z}_o$ on the frontier, given all other throughputs. The result is that these derivatives are now well defined, due to the piecewise linear nature of the DEA frontier. Furthermore, the derivatives can be characterized at given points by the ranges that they can take.

Rosen et *al.* give three approaches for the computation of marginal rates using linear programming techniques: minimum and maximum multiplier ratios, finite differences and a modified simplex tableau. Moreover, the authors show that the first two methods are equivalent - one is the dual linear program of the other. The simplex tableau method can generate two-dimensional sections of the efficient frontier and is an excellent tool for generating level plots for visualization purposes. For a thorough discussion of these three computational methods, the reader is referred to [ROSE95].

Figure 5-1 An Example of a Two-Dimensional Envelopment Surface



We will consider the finite differences approach in more detail herein. The first order partial derivatives of a continuous function $f(\mathbf{z})$ at a given point $\mathbf{z}_0$ can be approximated with the following:

$$\frac{\partial f}{\partial x_j}(\mathbf{z}_0) \approx \frac{f(\mathbf{z}_0 + h\mathbf{e}_j) - f(\mathbf{z}_0)}{h} \quad \text{(EQ 5-5)}$$

$$\frac{\partial f}{\partial x_j}(\mathbf{z}_0) \approx \frac{f(\mathbf{z}_0 - h\mathbf{e}_j) - f(\mathbf{z}_0)}{-h}\,. \quad \text{(EQ 5-6)}$$

where $\mathbf{e}_j$ is a vector of zeros with a one in the j-th position (unit vector), and h is a small finite number. (EQ 5-5) and (EQ 5-6) are similar in structure to the marginal rates from the left and right, although without the mathematical limiting process. As pointed out in [ROSE95], these first order approximations of partial derivatives give the exact slopes of DEA frontiers provided that h is small enough. This is because of the piecewise linear nature of the DEA frontier. One disadvantage of this method is that the solution may be sensitive to the choice of h.

The marginal rates at point $\mathbf{z}_0$ on the DEA frontier can be calculated using the finite differences approach by means of a three step procedure:

1. define a small increment h;
2. obtain z_{i0}' that results from increasing or decreasing the j-th throughput by h using the implicit function $z_{i0}\left(z_{10}, \ldots, z_{j0} \pm h, \ldots, z_{(m+s)0}\right)$;
3. compute the finite differences from z_{i0}, z_{j0} and h.

To obtain z_{i0} the following linear program is solved:

$$
\begin{array}{ll}
\underset{z_0', \lambda_0, \lambda}{\text{Max}} & z_{i0}' \\
\text{s.t.} & \mathbf{z}_0\lambda_0 + \mathbf{Z}\boldsymbol{\lambda} \geq \mathbf{z}_0' \\
& \lambda_0 + \mathbf{1}^T\boldsymbol{\lambda} = 1 \\
& \quad z_{l0}' = z_{l0} \quad , l \neq i, j \\
& \quad z_{j0}' = z_{j0} \pm h \\
& \lambda_0, \boldsymbol{\lambda} \geq \mathbf{0} \\
& \mathbf{z}_0' \geq \mathbf{0}
\end{array}
\qquad \text{(EQ 5-7)}
$$

This linear program solves for the i-th component of the throughput vector when the j-th component is increased or decreased by the small quantity h, such that the new point $\mathbf{z}_0'$ remains on the frontier. Once z_{i0}' in (EQ 5-7) is solved, the marginal rates can be calculated as follows:

$$
\mathrm{MR}_{ij}^{\pm} = \frac{z_{i0}' - z_{i0}}{\pm h} \qquad \text{(EQ 5-8)}
$$

As done in [ROSE95], it is possible to view the above treatment of marginal rates as a special case of the more general directional derivative. Extending the previous approach, we are interested in assessing the change in the throughputs in the direction of **v** that occurs with a small change in the throughput vector in the direction of **u** while remaining on the frontier. Furthermore, we denote as α the change in the **v**-direction that results from moving β units in the **u**-direction. These concepts can be more formally represented for the general (continuous) efficient frontier as $F(\mathbf{z}_0 + \alpha\mathbf{v} + \beta\mathbf{u}) = 0$ where the marginal rate is given by:

$$\mathrm{MR}_{\mathbf{u},\mathbf{v}}(\mathbf{z}) = \frac{\left|\partial F(\mathbf{z} + \beta\mathbf{u}) / \partial\beta\right|_{\beta = 0}}{\left|\partial F(\mathbf{z} + \alpha\mathbf{v}) / \partial\alpha\right|_{\alpha = 0}}. \qquad \text{(EQ 5-9)}$$

With a piecewise linear DEA frontier, these (directional) marginal rates can be defined to the right and to the left similar to (EQ 5-4).

5.3. Tradeoff Analysis Using DEA

The calculation of marginal rates at a given efficient point on a piece-wise linear frontier is limited to providing the change on one or more throughputs by modifying one or more different throughputs by *a small finite* change. Often, management requires the impact of modifying one or more throughputs by *much more* than those considered in calculating marginal rates. For instance, a project manager may want to assess the impact on project cost of increasing the size of the project by 400FP. Furthermore, marginal rates may vary greatly from point to point on the surface of the efficient frontier. Since marginal rates are local to the area on the frontier, one can't simply extrapolate the local marginal rate by the desired increment (or decrement), say 400FP, and assume that the marginal rates are constant throughout the increment (or decrement) and that the resulting point will remain on the efficient frontier. Thus, new methods for interactively exploring tradeoffs, with larger than small, finite changes, must be devised.

Let us first consider a method for exploring pairwise tradeoffs (*i.e.* obtain the *i*-th throughput that results from increasing or decreasing the *j*-th throughput by *h*). The finite differences approach of (EQ 5-7) can be adapted for this purpose. The major change necessary to adapt the approach would involve the user specifying any sized increment (or decrement) *h* such that $z_{j0}' = z_{j0} \pm h$ would not be beyond the realm of the production possibilities. One could then obtain z_{i0}' that results from increasing or decreasing the *j*-th throughput by *h* using an implicit function. Unlike the previous finite differences approach, no further calculation such as in (EQ 5-8) is necessary since obtaining z_{i0}' is the ultimate objective in this case.

The following linear program, adapted from (EQ 5-7), applies these ideas to calculate specified tradeoffs between throughput *i* and *j*:

$$\begin{aligned} \underset{\lambda, \lambda_0, \mathbf{z}_0', s}{\text{Max}} \quad & -s + \varepsilon\,(z_{i0}') \\ \text{s.t.} \quad & \mathbf{z}_0\lambda_0 + \mathbf{Z}\boldsymbol{\lambda} \geq \mathbf{z}_0' \\ & \lambda_0 + \mathbf{1}^T\boldsymbol{\lambda} = 1 \\ & z_{l0}' = z_{l0} \quad , l \neq i, j \\ & z_{j0}' + s = z_{j0} + h \\ & \lambda_0, \boldsymbol{\lambda} \geq \mathbf{0}\,,\ \mathbf{z}_0' \geq \mathbf{0}\,, \\ & \mathbf{s} \geq \mathbf{0} \text{ if } h > 0 \text{ and } \mathbf{s} \leq \mathbf{0} \text{ if } h < 0\,. \end{aligned} \qquad \text{(EQ 5-10)}$$

where ε is a non-archimedian infinitesimal. The linear program can be solved in two stages. The first stage minimizes a slack on the z_{j0}' variable in order to ensure that $z_{j0} \pm h$ does not go beyond the production possibility set. If it does, this slack will be non-zero. Then in the second stage z_{i0}' is maximized such that it is on the frontier, while holding the slack *s* at its minimum value.

Note that the j-th throughput is increased or decreased in an additive manner in (EQ 5-10). Alternatively, this can be modified such that j-th throughput is scaled:

$$
\begin{aligned}
\underset{\lambda,\lambda_0,\mathbf{z}_0',s}{\text{Max}} \quad & -s+\varepsilon\,(z_{i0}') \\
\text{s.t.} \quad & \mathbf{z}_0\lambda_0+\mathbf{Z}\boldsymbol{\lambda}\geq\mathbf{z}_0' \\
& \lambda_0+\mathbf{1}^T\boldsymbol{\lambda} = 1 \\
& z_{l0}' = z_{l0} \quad ,\, l\neq i,j \\
& z_{j0}'+s = z_{j0}\,(h) \\
& \lambda_0,\boldsymbol{\lambda}\geq\mathbf{0},\ \mathbf{z}_0'\geq\mathbf{0}, \\
& \text{if } h>1\text{ , } s\leq 0 \ \text{ if } j\in I \text{ and } s\geq 0 \ \text{ if } j\in O \\
& \text{if } h<1\text{ , } s\geq 0 \ \text{ if } j\in I \text{ and } s\leq 0 \ \text{ if } j\in O
\end{aligned}
\qquad \text{(EQ 5-11)}
$$

where I and O represent the set of input and output indices (as introduced in Section 2.2.1.). Note the change to the last equality constraint. This would allow an analysis to assess, for instance, the impact on the i-th throughput resulting from a 20% increase in the j-th throughput (where h=1.2), holding all other throughputs the same.

5.3.1. Some Generalizations

These two computational methods can also be extended to consider tradeoffs between more than two throughputs. For instance, a project manager may be interested in assessing the impact of a 10% increase in size on *both* project cost and duration, for the same level of quality. This can be accomplished by applying concepts similar to those of the directional derivative (see (EQ 5-9)).

For this purpose, it is useful to introduce the following notation. Let A represent the set of throughputs (*i.e.* throughput indices) which are changed by α as a result of increasing or decreasing the set of throughputs B by a specified amount β. Furthermore, let I represent the set of all inputs, O the set of all outputs and K

the set of throughputs such that $K \cap (A \cup B) = \emptyset$. Employing this notation, the linear program in (EQ 5-10) can be generalized to assess tradeoffs between two or more throughputs:

$$\max_{\lambda, \lambda_0, \mathbf{z}_0', \mathbf{s}}$$

$$\max_{\lambda, \lambda_0, \mathbf{z}_0', \mathbf{s}} \quad -\sum_j s_j + \varepsilon(\alpha) \ , \forall j \in B \tag{EQ 5-12}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{z}_0 \lambda_0 + \mathbf{Z}\boldsymbol{\lambda} \geq \mathbf{z}_0' \\ & \lambda_0 + \mathbf{1}^T \boldsymbol{\lambda} = 1 \\ & z_{j0}' + s_j = z_{j0} + \beta \ , \forall j \in B \\ & z_{i0}' - \alpha = z_{i0} \ , \forall i \in A \\ & z_{k0}' = z_{k0} \quad , \forall k \in K \\ & \lambda_0, \boldsymbol{\lambda} \geq \mathbf{0}, \ \mathbf{z}_0' \geq \mathbf{0} \ , \\ & \mathbf{s} \geq \mathbf{0} \text{ if } \beta > 0 \text{ and } \mathbf{s} \leq \mathbf{0} \text{ if } \beta < 0. \end{aligned}$$

There are now as many slacks as the number of elements in *B*. Furthermore, the elements *k* are the throughputs which are not contained in either set *A* or *B* and, thus, remain unchanged. Note that pairwise tradeoffs constitute a special case of the more general (EQ 5-12).

The program in (EQ 5-12) can be modified such that trade-offs in throughputs are calculated in a scalar fashion in the manner of (EQ 5-11):

$$\max_{\lambda, \lambda_0, \mathbf{z}_0', \mathbf{s}} \quad -\sum_j s_j + \varepsilon(\alpha) \ , \forall j \in B \tag{EQ 5-13}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{z}_0 \lambda_0 + \mathbf{Z}\boldsymbol{\lambda} \geq \mathbf{z}_0' \\ & \lambda_0 + \mathbf{1}^T \boldsymbol{\lambda} = 1 \\ & z_{j0}' + s_j = z_{j0}(\beta) \ , \forall j \in (B \cap O) \\ & z_{j0}' + s_j = z_{j0}(2 - \beta) \ , \forall j \in (B \cap I) \\ & z_{i0}' - z_{i0}(\alpha) = 0 \ , \forall i \in (A \cap O) \\ & z_{i0}' - z_{i0}(2 - \alpha) = 0 \ , \forall i \in (A \cap I) \\ & z_{k0}' = z_{k0} \quad , \forall k \in K \\ & \lambda_0, \boldsymbol{\lambda} \geq \mathbf{0}, \ \mathbf{z}_0' \geq \mathbf{0}, \end{aligned}$$

if $\beta > 1$, $s \geq 0$
if $\beta < 1$, $s \leq 0$.

Notice the change to the constraints containing α and β. The projection introduced in (EQ 4-14) is employed in this case since proportionate *increases* in outputs and proportionate *decreases* in inputs are desired (in the case of $\beta > 1$). If such a projection was not introduced and an input was a member of set *A*, then the resulting linear programming solution would be unbounded. This situation can occur since both inputs and outputs can belong to set *A*.

5.4. A Numerical Example

In order to illustrate the above concepts and computational methods, a simple two-dimensional example is provided. The data set represents an amalgamation of similar projects collected from two different Canadian Banks, and consists of the single input *Project Cost* and the single output *Project Size*. This example utilizes variable returns to scale (VRS) production technology. The data set is given in Table 5-1 on page 94 and is limited to two dimensions (2-D) in order that the efficient frontier and some sample steps of the tradeoff analysis can be illustrated with standard 2-D plots.

Table 5-2 on page 94 provides five sample iterations of tradeoff analysis. These iterations, along with the data set and efficient VRS frontier, are plotted in Figure 5-2 on page 95. The first iteration begins at DMU 17 on the efficient frontier with a cost of $194,000 and a size of 169FP. Suppose we are interested in assessing the impact on the project cost of increasing the size of the project by 300FP due to some major additions to the project scope. If we apply the computational approach given in (EQ 5-10), then *h*=300 and the starting point

$(z_{0i}, z_{0j}) = (194, 169)$ with variables z_1 and z_2 representing cost and size respectively. The size of the project is then increased from 169FP to 469FP and the corresponding project cost z_{0i}' is determined. In this case the new project cost is $459,100 resulting in an increase of $265,100 (for an additional 300FP). This is depicted in Figure 5-2 as well as in Table 5-2 as iteration 1. Notice that this iteration moves from one facet to another. Clearly, extrapolation of local marginal rates from the initial point (194, 169) by 300FP would result in a point outside of the production possibility set. This situation illustrates one of the fundamental reasons for developing the new methods presented herein.

Iteration 2 and 3 increment size by 400FP and 631FP respectively and remain on the same facet, as can be seen in Figure 5-2. Iteration 4, like the first iteration, moves off of one facet and on to a new facet. However, the last iteration, step 5, poses some potential problems. The main issue is that the size increment specified is 100FP, resulting in a new project size that is larger than the largest project (DMU 9) in the data set and is, thus, outside of the production possibility set $(1650 + 100 > 1716)$. However, the slack s in $z_{j0}' + s = z_{j0} \pm h$ eliminates this problem by ensuring that z_{j0}' is on the frontier and is feasible. In this case $z_{j0}' + s = 1716 + 34 = 1750$ and the slack s reduces the change in size by 34FP to the maximum size in the production possibility set of 1716FP. Thus, the slacks in (EQ 5-10) play an important role in ensuring feasibility, as well as improving usability, when computing tradeoffs.

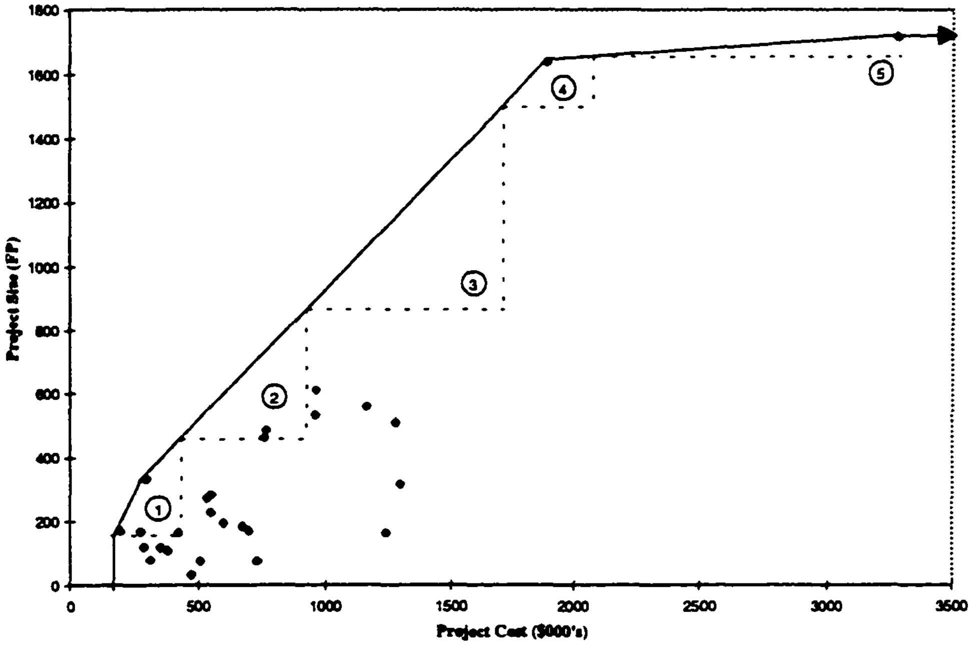
Table 5-1 Data Set of Software Projects

Project	Cost ($000's)	Size (FP's)	Project	Cost ($000's)	Size (FP's)	Project	Cost ($000's)	Size (FP's)
1	960	531	10	730	75	19	673	183
2	310	78	11	700	169	20	1279	509
3	550	227	12	1240	162	21	284	119
4	535	271	13	1162	557	22	507	74
5	273	168	14	766	485	23	294	334
6	550	282	15	378	108	24	348	118
7	1890	1639	16	421	165	25	598	194
8	1295	315	17	194	169	26	759	462
9	3283	1716	18	961	609	27	471	32

Table 5-2 Numerical Example of 2-D Trade-offs

Iteration	Starting Point (z_{0i}, z_{0j})	z_{0j} Increment (h)	New Point $(z_{0i}', z_{0j}+h-s)$	$(z_{0i}' - z_{0i})$	Slack (s)
1	(194, 169)	300	(459.1, 469)	265.1	0
2	(459.1, 469)	400	(948.3, 869)	489.2	0
3	(948.3, 869)	631	(1720, 1500)	771.7	0
4	(1720, 1500)	150	(2088.9, 1650)	368.9	0
5	(2088.9, 1650)	100	(3282.5, 1716)	1193.6	34

Figure 5-2 2-D Example of Tradeoff Analysis



Iterations 4 and 5 also demonstrate a weakness of this approach - a sensitivity of forecasts to outliers and "unrepresentative" frontiers. In this data set, project 9 lies on the frontier at the upper edge (see Figure 5-2). However, with more data collected for projects larger than project 9 (i.e. > 1716 FP), the forecasts produced from iterations 4 and 5 might be quite different. This type of sensitivity can occur particularly at the "edges" of the frontiers, and caution must be taken to interpret the results accordingly.

5.5. Generating Initial Forecasts

In the above example (Section 5.4.), the starting point for the first iteration was selected to be on the efficient frontier. Yet, in many situations such a "starting point" or an initial efficient forecast will not be known or available to project managers. Fortunately, the DEA frontier provides an excellent means for establishing these efficient forecasts for the second step of the above process.

For example, reconsider the fundamental project objectives of *cost*, *size*, *quality* and *duration* represented in the basic software production model of Figure 3-1 given in Section 3.2. For given estimates of size derived from FP or other size counts based on user requirements, and user specified quality and duration requirement (also considering available resources), it is possible to determine efficient estimates of project cost using the following linear program:

$$\begin{aligned} &\min_{x,\lambda} \; x \\ &\text{s.t.} \quad \mathbf{Y}\boldsymbol{\lambda} \geq \mathbf{y}_0 \\ &\qquad x - \mathbf{X}\boldsymbol{\lambda} \geq \mathbf{0} \\ &\qquad x, \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned} \tag{EQ 5-14}$$

(EQ 5-14) provides the minimum cost x of producing outputs $\mathbf{y}_0$ (in this case *size*, *quality* and *duration*) based on the efficient frontier constructed from the compiled software project data. It is also possible to provide these minimum cost estimates for other software production models presented in Chapter 3 that have a single cost input (such as those in Section 3.3.).

5.6. Applications to Project Management

Tradeoff analysis is typically conducted in an attempt to "balance" the various project objectives and constraints in order to develop the initial project plan. Furthermore, during the course of many projects, problems arise and many changes to the project scope occur necessitating ongoing and interactive tradeoff analysis. Initial project forecasts provide a critical starting point.

Forecasting and tradeoff analysis can be incorporated directly into the project management lifecycle and can be characterized by the following six steps:

1. Define and prioritize the project objectives
2. Generate an initial (efficient) project forecast
3. Generate alternatives using tradeoff analysis tools
4. Select the best alternative
5. Produce or revise the project plan
6. Monitor project and project environment (go to step 1 if necessary)

The first step in this process is a complete assessment or review of the main project objectives. These objectives and priorities are set based on the user requirements, available resources as well as many other environmental factors. Furthermore, the preference for tradeoffs amongst the objectives must be gauged, formally or informally, also considering these factors.

Generating the initial efficient forecast is discussed in Section 5.5.

Once the initial forecast has been obtained, the tradeoff analysis methods presented in Section 5.3. can be applied to generate and assess various alternatives. For instance, it may be necessary to reduce the size of the project in order to reduce the project duration, for the same cost and level of quality. The most appropriate alternative is then selected and the project plan is updated and revised.

The last step of this process is monitoring and controlling. Most projects have control systems that compare actual progress and results to the project plan. When problems arise, corrective action can be taken. Furthermore, the project environment must be scanned for important changes, such as changes in user requirements. Many times corrective action will require restarting the entire project management process and going back to step one.

5.7. Summary

This section presents several methods for the calculation of general tradeoffs adapted from the work of Rosen *et al.* [**ROSE95**] on the calculation of marginal rates. While information on marginal rates is of great value to management, marginal rates are limited to considering the impact of *infinitesimal* (or small finite) changes in one or more throughputs, on one or more other throughputs. Thus, marginal rates can be seen as a special type of tradeoff. However, the tradeoffs relevant to many project managers are more general, and often require assessing the impacts of changes ***much larger*** than those considered in marginal rates.

First, two methods are presented for the computation of pairwise tradeoffs: the impact on one throughput when another throughput is increased or decreased, with all other throughputs are the same. The first method considers additive changes, and the latter method allows for scalar changes. Furthermore, building on the concepts of directional derivatives, these two methods are extended to compute tradeoffs more general than basic pairwise tradeoffs. These more general methods assess the impact on one or more throughputs of the change in one or more of the other throughputs by a specified amount. Like the two pairwise models, the specified changes can be made in an additive or scalar manner.

While the methods for tradeoff analysis presented herein play an important part in the overall project management lifecycle, additional tools are needed to provide initial forecasts. Simple DEA models can be used to provide these initial project forecasts which also serve as a starting point for tradeoff analysis. In many cases, these forecasting and tradeoff analysis methods will be applied in an iterative manner within the project management cycle as problems or other issues necessitate constant revision to the project plan.

The main motivation for this work is to produce better forecasts and, therefore, better project plans, thereby reducing risk. More realistic and accurate project plans can help to reduce the risk of project failures as well as cost and schedule overruns [**ABDE91**]. Non-parametric frontier-based forecasting methods, such as DEA, produce forecasts based on best observed performance and do not impose many assumptions regarding the functional form of the production function. Both of these attributes can very advantageous and make DEA well suited to forecasting.

In the future, these methods could possibly incorporate probabilistic measures to represent uncertainty with respect to project input and output targets or outcomes. Furthermore, prices on inputs and outputs or preferences for tradeoffs could be incorporated directly into the analysis. It would be possible then to show the impact of a particular tradeoff on some profit or utility function. General methods to develop initial forecasts from DEA production possibility sets can also be developed. Finally, constraints on the allowable range of one or more inputs and outputs could easily be included in the models. These constraints could reflect restrictions on the project imposed by the environment, such as user requirements for project completion. The constraints would ensure that the forecasts satisfy such restrictions.

Parametric models (*c.f.* [**RAY91**]) can be constructed to provide estimates of project efficiency that can be used to moderate the best practice forecast once tradeoff analysis has been complete. Finally, the predictive power of the various software production models in Chapter 3 combined with the analytical techniques introduced in Chapter 6 could be tested by comparing actual to observed performance (see [**ABDE91**] for a discussion and caveats).

While these methods were developed to provide a forecasting tool for software engineering management, they are applicable to a wide variety of other domains. Furthermore, the general objectives of cost, size, quality and duration are also applicable to project management in other areas outside of software.

CHAPTER 6 *Summary and Conclusions*

"For too long, organizations have managed in spite of their information systems, rather than because of them."
Vaughan Merlyn & John Parkinson

6.1. Discussion and Summary

Software plays an increasingly important role in our economy. It forms a fundamental part of modern information technology (IT) which plays such a central role in organizational success and in sustaining competitive advantage. IT, and its crucial software component, enables flexible, leaner, more responsive organizations and allows for new types of organizational structures and workflows. Although several examples of excellence can be found, in general, our track record at producing software remains dismal (*c.f.* **[GIBB94]**). For example, a recent study by Ernst and Young reports that less than 10% of produced software systems are implemented with no significant modification, while 70% of systems are either scrapped before completion or never used even if completed **[MERL94]**.

Many experts argue that producing software must evolve beyond the category of a "craft" towards that of an engineering discipline, while continuously improving techniques and tools **[PRES92]**. Still others have noted the management problem associated with producing software as well as with evaluating and adopting newer technologies **[HUMP89]**. Fundamental to these various solution perspectives is objective measurement and analysis.

DEA is well suited for measuring the productive efficiency of multiple-input multiple-output production processes, thus, making it ideal for measuring and analyzing software production. Furthermore, previous applications in the literature have demonstrated how DEA provides a clearer and more consistent overall picture of performance than traditional techniques, such as performance ratios (*c.f.* [PARA95], [BANK86A], [SHER95]). DEA is also well suited as a management control tool since it can be applied for both planning and control purposes.

This thesis addresses several limitations encountered in the DEA literature with respect to its application to software engineering management. The first objective addresses modelling software production. Several applications of DEA to evaluating software producers have appeared in the literature (such as [BANK91B] and [PARA95]). However, in practice, these models are limited by their simplicity or by their limited applicability (or both).

The new software production models that are presented in Chapter 3 are designed with usefulness to practitioners in mind. These new models apply to both entirely new development and maintenance projects which consist of varying portions of new, modified and unmodified code. Incorporating these ideas, several models are developed that disaggregate the production process into several stages. This disaggregation process is crucial to evaluating new technologies since they usually target a particular phase or phases. For example, new test case generators and other testing tools target the independent testing phase of the production process, while new types of programming languages target the coding phase (and can be applied independently from the requirements analysis techniques utilized). Software production models are fundamental to any DEA analysis and, thus, are important to both the management planning and control processes.

Fundamental to the management control process is the assessment of organizational effectiveness. Such assessments provide crucial management insights into how well goals and objectives are achieved. Moreover, information regarding behavioral goals and objectives must be incorporated directly into the effectiveness analysis if quantitative measures are desired. DEA techniques can be applied for this purpose if multipliers are constrained to reflect realistic prices or other value measures, using the cone ratio DEA models. The value measures incorporated into the models can be, for instance, the relative priority of tradeoffs. Note that these models measure effectiveness relative to all other DMUs in the analysis.

Other techniques exist to measure a special case of effectiveness, known as overall efficiency, that have their origins in traditional economic production analysis (*c.f.* [**FARR57**]). If the market prices or value measures are known precisely, then these models can be applied to assess how far a particular DMU is from achieving a behavioral goal such as cost minimization, or revenue maximization relative to all other DMUs. Models also exist to measure overall profit efficiency, but some fundamental problems exist with these measures and their inherent projections.

The advantage of the measures of overall efficiency is that clear and rigorous definitions exist for what is being measured. Unfortunately, precise prices may not exist in many cases of management control, especially when the value measures are non-monetary. The situation with cone ratio models is the opposite: it is not clear what is being measured (in terms of defined measures of overall efficiency) but imprecise value measures can easily be incorporated into the models in the form of DEA multiplier constraints.

To address these problems, we begin by investigating the relationships between the various models. It is formally shown that as multiplier constraints tighten (i.e. get more and more restrictive), input and output-oriented cone ratio models converge to a measure of overall cost and a measure of revenue efficiency, respectively. The equivalence of measures of overall profit efficiency and input-oriented linked-cone DEA models is also shown. The limitations of these profit efficiency models are also highlighted. A new cone-ratio model to measure overall profit efficiency is introduced to overcome these limitations (EQ 4-14) as well as another model appropriate for effectiveness measurement using separable cones (EQ 4-16).

Throughout Chapter 4, it is stressed that the behavioral goal of the analysis and means of projection (determined by the efficiency measure) to the frontier must be consistent. Accordingly, a framework is presented which prescribes the appropriate application of the various existing and newly introduced models to the measurement of overall efficiency. This framework is also applicable to general effectiveness measurements using non-monetary and imprecise value measures. It is argued that this new model for measuring overall profit efficiency (EQ 4-14) is the most appropriate for this purpose since most organizational objectives involve both inputs and outputs. Although this work is motivated by the need to improve software engineering management, both the framework and methods are applicable to any domain appropriate for DEA usage.

Traditionally, DEA has been applied predominantly for management control purposes [**COOP94**]. However, some DEA theory and applications have been directed towards planning and prediction (*c.f.* [**BARR94**], [**GOLA93**]) as well as the calculation of marginal rates (a special case of tradeoffs) which can be useful for planning and decision-making [**ROSE95**]. Essential to software project planning is forecasting and tradeoff analysis. Tradeoff analysis is conducted by project managers to balance and tradeoff project objectives on an ongoing basis in response to scope changes and other unexpected difficulties.

The methods presented by Rosen et al. [**ROSE95**] are limited to computing marginal rates which assess the impacts of single unit changes of one or more throughputs (inputs or outputs) on one or more other throughputs. Chapter 5 generalizes and adapts the finite differences approach to calculating marginal rates for the purposes of general tradeoff analysis. Inherent in the results of tradeoff analysis is a new efficient forecast which can be used as the basis for the project plan. Like the material in Chapter 4, these methods are applicable to tradeoff analysis for general project management, and are not restricted only to software project planning.

6.2. Areas for Future Work

To gain a better understanding of the applicability of the multi-stage production models presented in Chapter 3, the models must be applied to real data sets. In this manner, further insights regarding adding or removing variables as well as strengths and limitations of the models can be gained. Applying a model to real data is an important step in the development of DEA production models which can not be, as yet, formally validated such as those to predict bank failures (see [**Barr94**]).

Different levels of engineering rigor can be incorporated directly into the models. One way of achieving this is to include a categorical variable into the model representing different levels of rigor, provided that an appropriate classification scheme is available. Several production models can also be developed that recognize and are appropriate for different levels of process maturity such as the five levels of the SEI Capability Maturity Model [**Hump89**].

Future work in the application of DEA to measure overall efficiency and effectiveness analysis will involve work in several areas. Measures of the tightness of cones could be developed or adapted from other areas in the literature. Combinations that are not included in the framework presented in Chapter 4 are not necessarily incorrect for application purposes, but simply do not conform to any well defined measures of overall efficiency. Thus, different combinations of cones and projections can be examined and their characteristics better understood. Finally, in terms of application by practitioners, better connections to the area of organizational theory can be made. While proper measures are important, they surely are not enough and additional guidance on the implementation of these methods such as in the management of change would be beneficial.

Future work in the area of tradeoff analysis will investigate the incorporation of uncertainty of outcomes (or objectives) directly into the analysis. Furthermore, factor prices or other value measures could also be incorporated into the analysis allowing the impact of a particular tradeoff on a value or profit function to be assessed. Often bounds exist on the acceptable range of inputs and outputs for a particular project. Constraints representing these bounds could be appended to the models and included in any user interfaces. The predictive power of the various software production models in Chapter 3, combined with the analytical techniques introduced in Chapter 6, could be tested by comparing actual to observed performance (see **[ABDE91]** for a discussion and caveats on validating forecasting tools). Finally, parametric models (*c.f.* **[RAY91]**) can be constructed to provide estimates of project efficiency that can be used to temper the best practice forecast once tradeoff analysis has been completed.

6.3. Summary of Contributions

- A new single stage software production model is presented that builds on that of **[REES93]** and **[PARA95]** but is significantly more general by incorporating new factors (Section 3.3.). Notably, size is measured as new and modified code, relative to the total system size which is a nondiscretionary output.

- A new two stage production model is presented that disaggregates the software production process into two stages: analysis, design, and coding in the first stage and testing in the second stage (Section 3.4.). This model is useful for evaluating the independent testing organization.

- A three stage model has also been introduced that divides the first stage of the previous model into two (Section 3.4.). The first stage comprises analysis and design, the second coding and the third testing. Multiple stages allow one to isolate the effects of technologies on particular phases of software production.

- It has been formally shown that as an input (price) cone of an input-oriented cone ratio model tightens, this model converges to measure overall cost efficiency (Section 4.2.).

- Similarly, it has been formally shown that as an output (price) cone of an output-oriented cone ratio model tightens, this model converges to measure overall revenue efficiency (Section 4.2.).

- It has also been shown that an input-oriented linked (price) cone ratio model is equivalent to Banker and Maindiratta's measure of overall profit efficiency, when all DMUs are evaluated using the same set of fixed market prices (Section 4.2.).

- A new type of projection has been introduced that, when incorporated into a linked-cone DEA model, is well suited for measuring overall profit efficiency. This projection proportionately increases outputs and decreases inputs (EQ 4-14) and (EQ 4-15). A ratio-like DEA model with no multiplier constraints has also been derived from this model that provides equivalent measures of profit efficiency (EQ 4-16).

- Another new separable cone DEA model has been presented that simultaneously minimizes inputs and maximizes outputs, but not necessarily proportionately (EQ 4-18). This model is designed to be compatible with behavioral goals consistent with separable cones. Similar to the above mentioned model, another ratio-like version of the model has been derived that contains no multiplier constraints (EQ 4-17).

- A framework has been constructed which prescribes the appropriate application of the various new and existing models (including cone ratio models) to measure overall efficiency (Section 4.4.). It has also been discussed how this framework can be applied more generally to measure effectiveness of achieving non-monetary behavioral goals.

- Two new models have been developed that are appropriate for conducting pairwise tradeoff analysis. One model calculates additive tradeoffs (EQ 5-10) and the other scalar tradeoffs (EQ 5-11).

- Two other more general models are introduced that can calculate tradeoffs amongst two or more throughputs. This follows concepts similar to the directional derivative. Like the previous two models, one calculates additive tradeoffs (EQ 5-12) and the second scalar tradeoffs (EQ 5-13).

- A model has been presented that can be used to generate efficient forecasts for software projects (EQ 5-14). This model can be applied to any software production model with a single input, such as those presented in Section 3.2. and Section 3.3.

References

[ABDE91] Abdel-Hamid, T. and Madnick, S. (1991), *Software Project Dynamics: An Integrated Approach*, Englewood Cliffs, NJ: Prentice Hall.

[ALBR79] Albrecht, A.J. (1979), "Measuring Application Development Productivity", *IBM Application Development Joint Share/Guide Symp.*, Monterey, CA.

[ALI94] Ali, A.I. (1994), "Computational Aspects of DEA", *DEA: Theory, Methodology and Applications*, Boston: Kluwer Academic Publishers, pp.63-88.

[ALI91] Ali, A.I., Cook, W.D., and Seiford, L.M. (19910, " Strict vs. Weak ordinal realtions for Multipliers in Data Envelopment Analysis", *Management Science*, 37, 6, pp.733-738.

[ALI95] Ali, A.I. and Lerme, C.S. (1995), "Comparative Advantage and Disadvantage in DEA," forthcoming in *Annals of Operations Research.*

[ALI93] Ali, A.I., and Seiford, L.M. (1993), "The Mathematical Programming Approach to Efficiency Analysis", *The Measurement of Productive Efficiency*, New York: Oxford University Press, pp. 120-159.

[ANTH89] Anthony, R.N., Dearden, J., and Bedford, N.M. (1989), *Management Control Systems Sixth Ed.*, Homewood, IL: Irwin, pp.185-186.

[BANK91A] Banker, R.D. (1991), "Stochastic DEA", *Carlson School of Management Working Paper, University of Minnesota*, Minneapolis, MN.

[BANK86A] Banker, R.D., Conrad, R.F., and Strauss, R.P. (1986), "A Comparative Application of DEA and Translog Methods: An Illustrative Study of Hospital Production", *Management Science*, 32, 1, pp.30-44.

[BANK91B] Banker, R.D., Datar, S.M., and Kemerer, C.F. (1991), "A Model to Evaluate the Variables Impacting the Productivity of Software Maintenance Projects", *Manaement Science*, 37, 1.

[BANK84] Banker, R.D., Charnes, A., and Cooper, W.W. (1984), "Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis", *European Journal of Operational Research*, 30, 9, pp.1078-1092.

[BANK88] Banker, R.D., and Maindiratta, A. (1988), "Nonparametric Anlaysis of Technical and Allocative Efficiencies in Production", *Econometrica*, 56, 6, *pp.*1315-1332.

[BANK86B] Banker, R.D., and Morey, R.C. (1986), "Efficiency Analysis for Exogenously Fixed Inputs and Ouputs," *Operations Research*, 34, 4, *pp.* 513-521.

[BANK93] Banker, R.D., and Morey, R.C. (1993), "Integrated System Design and Operational Decisions for Service Sector Outlets", *Journal of Operations Management*, 11, pp.81-98.

[BANK92] Banker, R.D., and Thrall, R.M. (1992), "Estimation of Returns to Scale Using Data Envelopment Analysis", *European Journal of Operational Reserach*, 62, pp.74-84.

[BARR94] Barr, R.S., Seiford, L.M., and Siems, T.F. (1994), "Forecasting Bank Failure: A Non-Parametric Frontier Estimation Approach," *Recherches Economique de Louvain*, 60, 4, *pp.*417-429.

[BARR97] Barr, Richard S. and Durcholz, M.L. (1997), "Parallel and Hierarchical Decomposition Approaches for Solving Large-Scale DEA Models," forthcoming in *Annals of Operations Research.*

[BAZA90] Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D. (1990), *Linear Programming and Network Flows*, New York: John Wiley and Sons.

[BOEH78] Boehm, B.W., Brown, J.R., Kaspar, J.R., Lipow, M., MacCleod, G.J., and Merrit, M.J. (1978), *Characteristics of Software Quality*, North Holland.

[BOEH81] Boehm, B.W. (1981), *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall.

[CHAR90] Charnes, A., Cooper, W.W., Huang, Z.M., and Sun, D.B. (1990), "Polyhedral Cone-Ratio DEA Models with an Illustrative Application to Large Commerical Banks", *Journal of Econometrics*, 40, pp.73-91.

[CHAR94] Charnes, A., Cooper, W.W., Lewin, A.Y., and Seiford, L.M. eds. (1994), *Data Envelopment Analysis: Theory Methodology and Applications*, Norwell, Massachusetts: Kluwer Academic Publishers.

[CHAR78] Charnes, A., Cooper, W.W., and Rhodes, E. (1978), "Measuring the Efficiency of Decision Making Units", *European Journal of Operational Research*, 2, 6, pp.429-444.

[CHAR89] Charnes, A., Cooper, W.W., Wei, Q.L., and Huang, Z.M. (1989), "Cone-Ratio Data Envelopment Analysis and Multi-Objective Programming", *International Journal Systems Sciences*, 20, pp.1099-1118.

[CHAV94] Chavas, J-P, and Cox, T. (1994), "A Primal-Dual Approach to Nonparametric Productivity Analysis: The Case of U.S. Agriculture", *The Journal of Productivity Analysis*, 5, pp.359-373.

[CONT86] Conte, S.D., Dunsmore, H.E., and Shen, V.E. (1986), *Software Engineering Metrics and Models*, Menlo Park, CA: Benjamin/ Cummings Pub. Co.

[COOP94] Cooper, W.W., Sueyoshi, H. and Tone, K. (1994), "DEA: Survey and Interpretations," *Communications of the Operations Research Society of Japan*, October 1994.

[DOYL94] Doyle, J.R. and Green, R.H. (1994), "Cross-Evaluation in DEA: Improving Discrimination Among DMUs", *INFOR*, 33, 3, *pp*.205-222.

[DRUC77] Drucker, P. (1977), *An Introductory View of Management*, New York: Harper College Press.

[ELAM91] Elam, J. (1991), "Evaluating the Efficiency of IS Organizations using Data Envelopment Analysis", *Proc. of International Function Point Users Group 1990 Fall Conference*, October 1991.

[FÄRE86] Färe, R., Grosskopf, S., and Lovell, C.A.K. (1986), *The Measurement of Efficiency of Production*, Boston: Kluwer-Nilhoff Publishing.

[FÄRE88] Färe, R. (1988), *Fundamentals of Production Theory*, Berlin: Springer-Verlag.

[FÄRE94] Färe, R., Grosskopf, S. (1994), *Cost and Revenue Constrained Production*, New York: Springer-Verlag.

[FARR57] Farrel, M.J. (1957), "The Measurement of Productive Efficiency", *Joural of the Royal Statistical Society*, 120, 3, pp.253-281.

[FERR94] Ferrier, G.D. (1994), "Assessing the Relative Performance of Hospitals", Presented at *TIMS XXXII*, Anchorage , Alaska, June 12-15, 1994.

[FENT91] Fenton, N.E. (1991), *Software Metrics: A Rigorous Approach*, London: Chapman and Hall.

[FENT94] Fenton, N.E. (1994), "Software Measurement: A Necessary Scientific Basis", *IEEE Transactions on Software Engineering*, 20, 3.

[GIBB94] Gibbs, W.W. (1994), "Software's Chronic Crisis", *Scientific American*, September 1994, pp.86-95.

[GOLA88] Golany, B. (1988), "An Interactive MOLP Procedure for the Extension of DEA to Effectiveness Analysis", *Journal of the Operational Research Society*, 39, 8, pp.725-734.

[GOLA93] Golany, B., Phillips, F.Y., and Rousseau, J.J. (1993), "Models for Improved Effectiveness Based on DEA Efficiency Results", *IIE Transactions*, 25, 6, pp.2-10.

[GRAD87] Grady, R.B., and Caswell, D.L. (1987), *Software Metrics: Establishing a Company Wide Program*, Prentice-Hall.

[GRIF87] Griffin, R.W. (1987), *Management* 2nd Edition, Boston: Houghton Mifflin Co.

[GROS93] Grosskopf, S. (1993), "Efficiency and Productivity", in *The Measurement of Productive Efficiency: Techniques and Applications*, Fried et al. Editors, NY: Oxford University Press.

[HALL91] Hall, R.H. (1991), *Organizations: Structures, Processes, Outcomes, 5th Edition*, New Jersey: Prentice Hall.

[HUMP89] Humphrey, W. (1989), *Managing the Software Process*, Reading, MA: Addison Wesley.

[JEFF90] Jefferry, D.R. and Low, G. (1990), "Generic Estimation Tools in Management of Software Development," *Software Engineering Journal*, 5, 4.

[JONE86] Jones, C. (1986), *Progamming Productivity*, New York: McGraw-Hill Book Company.

[JONE91] Jones, C. (1991), *Applied Software Measurement*, New York: McGraw-Hill Book Company.

[KEEN76] Keeney, R.L., and Raiffa, H. (1976), *Decisions with Multiple Objectives*, New York: John Wiley and Sons.

[KEME87] Kemerer, C.F. (1987), "An Emprical Validation of Software Cost Estimation Models", *Communications of the ACM*, 30, 5.

[KERZ95] Kerzner, H. (1995), *Project Management: A Systems Approach to Planning, Scheduling and Controlling*, New York: Van Nostrand Reinhold.

[KORN91] Kornbluth, J.S.H. (1991), "Analysing Policy Effectiveness Using Cone Restricted Data Envelopment Analysis", *Journal of the Operational Research Society*, 42, 12, pp.1097-1104.

[LOVE93] Lovell, C.A.K (1993), "Production Frontiers and Productive Efficiency, *The Measurement of Productive Efficiency*, New York: Oxford University Press, pp.3-67.

[MCCA77] McCall, J., Richards, P., and Walters, G. (1977), "Factors in Software Quality", three volumes, NTIS AD-A049-014, 015, November 1977.

[MERL94] Merlyn, V. and Parkinson, J. (1994), *Development Effectiveness: Strategies for IS Organizational Transition*, New York: John Wiley & Sons Inc.

[PARA95] Paradi, J.C., Reese, D.N., Rosen, D. (1995), "Applications of DEA to Measure the Efficiency of Software Production at Two Large Canadian Banks", in press the *Annals of OR*.

[PAUL95] Paulk, M., Weber, C., Curtis, B., Chrissis, M.B. *et al.* (1995), *The Capability Maturity Model: Guidelines for Improving the Software Process*, Reading, MA: Addision Wesley.

[PRES92] Pressman, R.S. (1992), *Software Engineering: A Practitioner's Approach*, New York: McGraw-Hill Inc.

[PUTN92] Putnam, L.H., and Myers, W. (1992), *Measures for Excellence: Reliable Software on Time, Within Budget*, New Jersey: Yourdon Press.

[RAY91] Ray, S.C. (1991), "Resource-Use Efficiency in Public Schools: A Study of Connecticut Data," *Management Science*, 37, 12, *pp.* 1620-1628.

[REES93] Reese, D.N. (1993), *Applications of DEA to Measure the Efficiency of Software Production in the Financial Services*, M.A.Sc. Thesis, Department of Industrial Engineering, Universtiy of Toronto.

[RETZ92] Retzlaff-Roberts,D.L., Morey,R.C. (1992), "A Comparison of Stochastic Allocative DEA Methods", Presented at EURO/TIMS XXXI, Helsinki, Finland, June 1992.

[ROLL91] Roll, Y., Cook, W., and Golany, B. (1991), "Contolling Factor Weights in Data Envelopment Analysis", *IEE Transactions*, 23, 1, pp.2-9.

[ROLL93] Roll, Y., and Golany, B. (1993), "Alternate Methods of Treating Factor Weights in DEA", *Omega*, 21 ,1, pp.99-109.

[ROSE95] Rosen, D., Schaffnit, C. and Paradi, J.C. (1995), "Marginal Rates and Two-Dimensional Level Curves in DEA", forthcoming in the *Journal of Productivity Analysis.*

[SAGE95] Sage, A.P. (1995), *Systems Management for Information Technology and Software Engineering*, New York: John Wiley & Sons.

[SEIF96] Seiford, L.M. (1996), "DEA: The Evoluation of the State of the Art (1978-1995)," *The Journal of Productivity Analysis*, 7, *pp*.99-137.

[SEIF90] Seiford, L.M., and Thrall, R.M. (1990), "Recent Developments in DEA: The Mathematical Programming Approach to Frontier Analysis," *Journal of Econometrics*, 46.

[SENG89] Sengupta, J.K. (1989), *Efficiency Analysis by Production Frontiers: The Non-Parametric Approach*, Boston: Kluwer Academic Publishers.

[SHER95] Sherman, H.D., and Ladino, G. (1995), "Managing Bank Productivity Using DEA", *Interfaces*, 25, 2, pp.60-73 .

[SIEM92] Siems, T.F. (1992), "Quantifying Management's Role in Bank Survival," *Federal Reserve Bank of Dallas Economic Review*, Jan. 1992, *pp*. 29-41.

[SUEY92] Sueyoshi, T. (1992), "Measuring Technical, Allocative and Overall Efficiencies Using a DEA Algorithm", *Journal of the Operational Research Society*, 43, 2, pp. 141-155.

[TAMU76] Tamura, K.A. (1976), "A Method for Construction the Polar Cone of a Polyhedral Cone with Applications to Linear Multicriteria Decision Problems", *Journal of Optimization Theory Appl.*, 19, pp.547-564.

[THOM90] Thompson, R.G., Langemeier, L.N., Lee, C.T., Lee, E., and Thrall, R.M. (1990), " The Role of Multiplier Bounds in Efficiency

Analysis with Application to Kansas Farming", *Journal of Econometrics*, 46, pp.93-108.

[THOM92] Thompson, R.G., Dharmapala, P.S., Grosskopf, S., and Thrall, R.M. (1992), "DEA Linked-Cone Efficiency and Profitability with Applications to Illinois Coal Mines", *Presented at EURO XII/TIMS XXXI Joint International Conference, OR/MS, Helsinki, Finland, June 1992.*

[THOM94] Thompson, R.G., Dharmapala, P.S., Rothenburg, L.J. and Thrall, R.M. (1994), "DEA ARs and CRs Applied to Worldwide Major Oil Companies", *The Journal of Productivity Analysis*, 5, 2, pp.181-203.

[VARI84] Varian, H. (1984), "The Nonparametric Approach to Production Analysis", Econometrica, 54, pp.579-597.

[WONG90] Wong, Y-H.B., and Beasley, J.E.. (1990), "Restricting Weight Flexibility in DEA", *Journal of the Operational Research Society*, 41, 9, pp.829-835.

[WORT94] Wortman, D.B. (1994), "Software Engineering", CSC 2105 Lecture Notes, Department of Computer Science, University of Toronto.

[YOUR96] Yourdon, E. (1996), *Rise and Resurrection of the American Programmer*, Upper Saddle River, New Jersey: Yourdon Press.

[YU85] Yu, P-L. (1985), *Multiple Criteria Decision Making: Concepts, Techniques and Extensions*, New York: Plenum Press.

Glossary

Additive Model	Measures efficiency in a non-radial manner as a sum of input and output slacks.
Allocative Efficiency	The ability to combine inputs and outputs in optimal proportions in the presence of market (or other) prices.
Bug / Defect	A software error that produces incorrect or undesirable outcomes.
cone	Two types of cones are relevant herein. A *convex* cone is a series of rays emanating from the origin. A *polyhedral* cone is the intersection of a finite number of half-spaces whose hyperplanes pass through the origin.
CRS	Constant returns to scale. Proportionate increases in inputs results in the same proportionate increase in outputs.
CRS Frontier	A production frontier which exhibits CRS.
DEA	Nonparametric, linear programming methods to applied to efficiency measurement which require no *a priori* specification of functional form of the frontier or weights is required.
Development	The production of new software (different from *maintenance*).
DMU	Decision Making Unit. A unit included in the analysis.
DRS	Decreasing returns to scale. Proportionate increases in inputs results in a proportionately smaller increase in outputs.
Efficiency	A general term often associated with performing activities as well as possible.
Effectiveness	How well a DMU is achieving its objectives relative to all other DMUs in analysis. *Overall efficiency* can be seen as a special case of *effectiveness*.

Efficient Frontier A frontier or surface (usually piece-wise linear in DEA) determined from the best observed or best practice production.

Envelopment Surface See *Efficient Frontier.*

Environmental Variable A non-discretionary variable that indirectly affects the efficiency score by influencing the selection of efficient peers.

FP function point: a measure of software size

graph The set of points where $y = f(\mathbf{x})$

hull the closure of a set under some operation. A closure is the smallest closed set containing a given set.

hyperplane a supporting hyperplane that constrains a particular facet of the efficient frontier:

$$\sum_{i=1}^{s} \mu_i * y_i - \sum_{j=1}^{m} v_j * x_j + \omega = 0$$

Input-Oriented Model A DEA model that measures efficiency in terms of a proportionate reduction of inputs.

IRS Increasing returns to scale. Proportionate increases in inputs results in a proportionately larger increase in outputs.

IS Information Systems.

LOC Lines of code: a measure of software size.

Maintenance Enhancements, repairs and other work conducted on existing software.

Management Control The process of ensuring the implementation of plans.

Management Planning The process of determining "what to do".

Marginal Rate A trade-off between various inputs and outputs. Mathematically, it is a partial derivative on the frontier.

Multiplier Cone Multiplier restrictions in the form of a polyhedral cone.

Netput See *Throughput.*

Output-oriented Model A DEA model that measures efficiency in terms of a proportionate increase in outputs.

Overall Efficient Both technically and allocatively efficient. Calculated as the product of technical and allocative efficiency.

Peer Group The set of efficient units against which inefficient units are compared. The efficient targets are a linear (and sometimes convex) combination of these peers.

Performance Ratio Usually a ratio of output over input. Implicitly assumes CRS.

Polar Cone A polar cone C^0 of a given cone C is defined as follows:

$$\mathbf{w}^* \in C^0 = \{\mathbf{w}^* \mid \mathbf{w}^T\mathbf{w}^* \le 0, \forall \mathbf{w} \in C\}.$$

Productive Efficiency Comparisons of actual producer performance to best-practice performance.

Production Frontier See *Production Function.*

Production Function A function that indicates the outputs that an organization produces for a specified combination of inputs. Furthermore, these functions specify the maximum output feasible for a given set of inputs in a *technically efficient* manner.

Production Technology See *Production Function.*

Software Engineering The discipline of producing software.

SLOC Source lines of code: a measure of software size. See *LOC.*

Technical Efficiency The ability to produce as much output as feasible for a given set of inputs.

Tightness "*Tightness*" of a cone. This refers to the degree of multiplier flexibility ranging from complete flexibility (as in measures of technical efficiency) to fixed prices (as in measures of overall efficiency).

Trade-off The impact on one on or more throughputs by changing one or more other throughputs. See also *Marginal Rate.*

Throughput A *variable*, either an input or an output. A *throughput* vector contains all inputs and outputs.

Variable An input or an output (including environmental variables) included in a DEA production model.

VRS Frontier	A variable returns to scale frontier which allows for constant, increasing and decreasing returns to scale (CRS, IRS and DRS, respectively).

Nomenclature

$\mathbf{A} = [\mathbf{A}^o, \mathbf{A}^i]$	$k \times (m+s)$ matrix of coefficients for k multiplier constraints (intersection form)
class A	contains many production possibility sets that meet specific criteria
A	set of throughputs modified by α units (specified by direction $\mathbf{v}$)
B	set of throughputs modified by β units (specified by direction $\mathbf{u}$)
$\mathbf{B} = [\mathbf{B}^o, \mathbf{B}^i]$	A sum form cone constraint matrix (where the cone is defined by a linear combinations of a finite number of extreme vectors)
$\mathbf{c} \geq \mathbf{0}$	a vector of input prices
D	the subset of discretionary variables
$D_L(\mathbf{x}_j, \mathbf{y}_j)$	input distance function
E, E'	the set of DMUs which maximize profit
$F_L(\mathbf{x}_j, \mathbf{y}_j)$	output distance function
h	an additive increment or scalar factor specified when calculating marginal rates and trade-offs.
I	the set of input variables
Int( )	the set of all interior points in a set (*i.e.* the largest open subset in the set)
L	the outer bound of class A
$L(\mathbf{y})$	represents the production technology as an input set (*i.e.* the set of feasible combinations of inputs associated with producing output vector $\mathbf{y}$)

$L'(\mathbf{c}, \mathbf{y})$	represents the production technology as an input cost set (*i.e.* the set of feasible costs associated with producing output vector **y** for a given cost vector **c**)
m	the number of inputs
$MR_{ij}(z_0)$	the marginal rate of throughput i to throughput j at the point z_0 on the frontier
n	the number of DMUs
$num(\ )$	the number of elements in a set
N	the subset of nondiscretionary variables
O	the set of output variables
OE_i	overall cost efficiency
OE_o	overall revenue efficiency
OE_{io}	overall profit efficiency
$\mathbf{p}$	$\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T)$
$\mathbf{r} \geq \mathbf{0}$	a vector of output prices
s	the number of outputs
$\pm s_j$	a linear programming slack for throughput j
S	the inner bound of A
T	the production possibility set
$\mathbf{u}$	directional vector used to calculate directional derivatives and general trade-offs
U	an output cone
$\mathbf{v}$	directional vector used to calculate directional derivatives and general trade-offs
V	an input cone
$\mathbf{w}^T = (\mu^T, \mathbf{v}^T)$	a vector of all input and output multipliers
W	a general multipliers cone
$\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})^T$	a vector of m inputs
$\mathbf{y}_j = (y_{1j}, \ldots, y_{sj})^T$	a vector of s outputs

$(\mathbf{x}_0, \mathbf{y}_0)$	input and output vector for a particular DMU_0
$\mathbf{X}$	$m \times n$ matrix of inputs
$\mathbf{Y}$	$s \times n$ matrix of outputs
z	$z^T = (y^T, -x^T)$ a throughput or netput vector
$\mathbf{Z}$	$(m+s) \times n$ matrix of throughputs
α	change in **v** direction as a result of moving β units in the **u** direction (used in calculating directional derivatives)
β	specified increment in the **u** direction used in calculating directional derivatives
ε	non-archimedian infinitessimal
λ_j	non-negative scalar that can indicate the peer group
θ	input-oriented technical efficiency score
$\nu = (\nu_1, \nu_2, \ldots, \nu_m)^T$	vector of input weights or multipliers
$\mu = (\mu_1, \mu_2, \ldots, \mu_s)^T$	vector of output weights or multipliers
π	a measure of overall profit efficiency
ϕ	output-oriented technical efficiency score
$\chi^{*T} = (\mu^*, \nu^*)^T$	denotes the vector of optimal multipliers that describe a supporting hyperplane that constrains a facet of the frontier (note that the optimal multipliers may not be unique)

APPENDIX A *Chapter 3 Appendix*

A.1. Data set and Complete DEA Results

A.1.1. Basic Production Model Results

This data set was drawn from a large Canadian bank and contains thirteen completed projects. Four of the projects represent new development, while the remaining nine represent enhancement type maintenance (not simply bug fixes). This maintenance constitutes varying degrees of enhancements (new code added) as well as substantial portions of existing code that has been modified. Notably, six of the nine maintenance projects have sizeable portions of unmodified code. Unmodified code is found by subtracting the *new* and *modified* code totals from the *total system size*. Due to a small sample size the *new* and *modified* code variables have been aggregated and the *quality* measure has not been included due to incomplete and inconsistent data. The duration measure was transformed as discussed in Section 3.2. for analysis purposes. The following results list the raw, untransformed data.

The cost of each project is approximated by the total effort (labour). The total project cost is normally calculated at the Bank by multiplying the total effort by a fixed chargeout rate and adding any extraordinary items such as consultant fees etc. Only the total effort for each project was available from the cooperating Bank.

Table A-1 Efficiency Scores and Peer Groups for Basic Model

Project Number	Score	Peer Group Coefficients	Input	Outputs	
			Effort (Work Months)	New & Modified Size (FP)	Duration (Months)
1	0.37	6 5 0.195 0.805	96	531	21
2	1	2 1	31	78	7
3	1	3 1	55	227	5
4	0.55	6 5 0.055 0.945	53.5	271	23
5	1	5 1	27.25	168	16
6	1	6 1	70.5	2032	13
7	0.59	6 2 5 0.081 0.417 0.501	55	282	12
8	0.32	5 6 0.211 0.789	189	1639	14
9	0.25	6 2 5 0.093 0.302 0.604	129.5	315	13
10	0.19	5 6 0.17 0.83	325.25	1716	24
11	0.40	5 1 0.444 0.555	73	75	11
12	0.39	6 5 0.001 0.999	70	169	36
13	0.22	5 1	124	162	28

A.1.2. Enhanced Production Model Results

Table A-2 Efficiency Scores and Peer Groups for Enhanced Model

Project Number	Score	Peer Group Coefficients	Input	Outputs		
			Effort (Work Months)	New & Modified Size (FP)	Duration (Months)	Total System Size (FP)
1	0.42	6 7 5 0.183 0.199 0.619	96	531	21	805
2	1	2 1	31	78	7	78
3	1	3 1	55	227	5	227
4	0.55	6 5 0.055 0.945	53.5	271	23	271
5	1	5 1	27.25	168	16	168
6	1	6 1	70.5	2032	13	2032
7	1	7 1	55	282	12	1660
8	1	8 1	189	1639	14	5287
9	0.25	6 2 5 0.093 0.302 0.604	129.5	315	13	315
10	0.19	5 6 0.17 0.83	325.25	1716	24	1716
11	0.43	7 5 2 0.086 0.396 0.517	73	75	11	250
12	0.41	7 5 0.04 0.96	70	169	36	228
13	0.30	7 5 0.341 0.659	124	162	28	677

Note: Total System Size is an environmental or non-discretionary output.

APPENDIX B *Appendix to Chapter 4*

B.1. Feasibility of Cone-Restricted CCR & BCC Models

The feasibility of the cone-restricted CCR model was addressed in [**CHAR89**] and [**CHAR90**]. The authors show that input, output and separable multiplier cones, when applied to the CCR model, have at least one efficient DMU and, therefore, a feasible solution. While linked cone CCR models have been employed in the literature (*c.f.* [**THOM90**] and [**THOM92**]), one cannot be assured of a feasible solution. Infeasibility will occur when the conical production possibility set, built around the observed production data, does not 'intersect' with the linked multiplier constraints constructed from external prices (or other value measures).

Consider now an $(m + s)$ dimensional VRS production possibility set, for a BCC model, constructed using the axioms given in [**BANK84**]. The vectors normal to the facets of this production (*i.e.* efficient) frontier indicate the optimal multiplier values (note that the multipliers are not unique at all points on the frontier). Furthermore, the set of optimal multipliers can be represented by a polyhedral cone constructed using the extreme directions of the set of normal vectors (see [**BAZA90**]).

For example, the extreme normal directions of a two dimensional VRS production possibility set of throughputs $\mathbf{z}^T = \{y, -x\}$ would be $\{0, 1\}^T$ and $\{1, 0\}^T$. Similarly, the extreme directions of a three-dimensional VRS production possibility set (a total of three inputs and outputs) of throughputs would be: $\{0, 0, 1\}^T$, $\{0, 1, 0\}^T$and $\{1, 0, 0\}^T$.

One could formulate a cone representing the set *W* of optimal BCC multipliers in the following manner:

$$\mathbf{B}^T = \mathbf{B} = \mathbf{I} = \begin{bmatrix} 1 & 0 \ldots 0 & 0 \\ 0 & 1 \ldots 0 & 0 \\ \ldots & \ldots & \ldots \\ 0 & 0 \ldots 0 & 1 \end{bmatrix},$$

and $W = \{\mathbf{w} : \mathbf{Bw} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0}\}$, where **B** is an $(m+s) \times (m+s)$ matrix and **w** is an $(m+s) \times 1$ vector. Clearly $W = E^{m+s}$ and, thus, a BCC model with any linked multiplier cone $W' \subseteq E^{m+s}$ of full (m+s) dimension will have a feasible solution since $W' \cap W$ is non-empty.

Similar arguments could be constructed for BCC models with various cones of less than full dimension such as linked, separable, input or output cones. Moreover, these cone-restricted BCC model feasibility characteristics apply to any DEA models based on standard VRS production possibility sets as presented in [BANK84].

B.2. Duals of Selected Linear Programs

The following models are the duals of selected models presented in Chapter 4. However, these models are generalized by making the following substitutions for the price vector $\mathbf{p}^T = (\mathbf{r}^T, \mathbf{c}^T) : \mathbf{p}^T = \mathbf{B}$, where $\mathbf{Bw} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0}$. These substitutions were also discussed in Section 4.4.

The first model is the dual linear program of that described in (EQ 4-9).

$$\begin{array}{ll} \min\limits_{\mu, \nu, \omega} & \omega \\ \text{s.t.} & y_0^T\mu - x_0^T\nu = 1 \\ & \mathbf{Y}^T\mu - \mathbf{X}^T\nu - \omega\mathbf{1} \leq \mathbf{0} \\ & \mu, \nu \geq \mathbf{0} \end{array} \qquad \text{(EQ B-1)}$$

It is interesting to note that the "linkage" made between the inputs and outputs, in terms of the throughput augmentation factor, results in the both inputs and outputs "linked" in the normalization constraint. Clearly, $\omega \geq 1$ for this model.

However, in the (more general) dual of (EQ 4-18) given by (EQ B-2), the input and output normalization is not "linked" due to the separate radial input reduction and output augmentation factors. From the constraints it can be seen that $\omega \geq 0$, where 0 represents a DMU on the efficient frontier.

$$\begin{array}{ll} \min\limits_{\mu, \nu, \omega} & \omega \\ \text{s.t.} & y_0^T\mu = 1 \\ & x_0^T\nu = 1 \\ & \mathbf{Y}^T\mu - \mathbf{X}^T\nu - \omega\mathbf{1} \leq \mathbf{0} \\ & \mu, \nu \geq \mathbf{0} \end{array} \qquad \text{(EQ B-2)}$$

(EQ B-3) gives the dual of (EQ 4-15), the newly proposed model to measure overall profit efficiency. Similar to (EQ B-1), the normalization constraint involves both the inputs and the outputs. Furthermore, notice the similarity between the objective function of (EQ B-3) and the numerator of (EQ 4-16).

$$\begin{aligned} \underset{\mu, \nu, \omega}{\text{Min}} \quad & \omega - 2\nu^{T}\mathbf{x}_0 \\ \text{s.t.} \quad & \mathbf{y}_0^{T}\mu - \mathbf{x}_0^{T}\nu = 1 \\ & Y^{T}\mu - X^{T}\nu - \omega\mathbf{1} \le \mathbf{0} \\ & \mu, \nu \ge \mathbf{0} \end{aligned} \qquad \text{(EQ B-3)}$$

APPENDIX C *Additional Reference Material*

C.1. Relevant Unpublished References

"Applications of DEA to Measure the Efficiency of Software Production at Two Large Canadian Banks" [**PARA95**] has been accepted for publication in a special edition of the Annals of Operations Research. However, it has not yet appeared and, thus, is currently available only from the authors. It has been included in this document for the convenience of the reader.

This paper is especially relevant to Chapter 3.

Applications of DEA to Measure the Efficiency of Software Production at Two Large Canadian Banks

J.C. Paradi, D.N. Reese, D. Rosen

Centre for Management of Technology and Entrepreneurship
University of Toronto

Department of Industrial Engineering
4 Taddle Creek Road, Toronto, Ontario, M5S 1A4, Canada
Tel: (416) 978-6924
Fax: (416) 978-3877
email: paradi@ie.utoronto.ca

Feb. 18, 1995.

ABSTRACT

This paper presents two empirical studies of software production conducted at two large Canadian Banks. For this purpose, we introduce a new model of software production that considers more outputs than those previously cited in the literature. The first study analyses a group of software development projects and compares the ratio approach to performance measurement to the results of DEA. It is shown that the main deficiencies of the performance ratio method can be avoided with the latter. Two different approaches are employed to constrain the DEA multipliers with respect to *subjective managerial goals*. As is further shown, incorporating subjective values into efficiency measures must be done in a careful and rigorous manner, within a framework familiar to management. The second study investigates the effect of quality on software maintenance (enhancement) projects. Quality appears to have a significant impact on the efficiency and cost of software projects in the data set. We further show the problems that may result when quality is excluded from the production models for efficiency assessment. In particular, we show some of the misleading results that can be obtained when the simple, traditional, ratio definition of productivity is used for this purpose.

Key Words: Data Envelopment Analysis, Efficiency Measurement, Multiplier Constraints, Software Productivity, Software Quality.

Acknowledgments: The authors would like to thank Dr. Claire Schaffnit for many fruitful discussions and valuable comments on the manuscript. Finally, we would like to express our appreciation to Maris, Kyoko, Tony and Mike at the banks.

1. Introduction

Software production has become one of the major economic activities throughout the world. Currently, it is estimated that over $500 billion is spent annually on producing software worldwide (Wortman 1994). As the industry increases in size, so does the problem of managing software production. Reports of overruns of 100% to 200% are not uncommon in private sector software projects. For instance, Jones reported that the average U.S. software project runs a year late, costs twice as much as forecasted, and performs below user expectations (Jones 1991). On this problem, it has been noted that nearly three quarters of large systems are deemed operating failures, that either do not perform as intended or are scrapped (Gibbs 1994). This management problem is further aggravated by the fact that software systems double in size every five to ten years. Hence, substantial effort is currently being devoted to measure the performance of the software production process, in order to determine methodologies for its improvement.

New measures of software output are developed constantly to help address the software management problem. However, as pointed out by some authors, much of the published work on software metrics is theoretically flawed (*c.f.* Fenton 1994). Software Managers are particularly interested in the measurement of *external attributes* of the process such as *reliability, productivity,* and *quality.* In particular, productivity is commonly defined, in an oversimplistic way, as a ratio of the size of the code delivered and the effort expended. Quality is further defined as the ratio of software defects discovered during testing and the size of the project.

Of further interest to management is the effect of quality, and quality assurance, on software project cost and on programmer productivity. Some researchers have found that the quality assurance effort, which includes software testing and inspection, has a strong nonlinear effect on costs (Abdel-Hamid *et al.* 1991). For a particular test case, these authors found that the quality assurance effort could affect total project cost by as much as 33%. Boehm (1981) further concludes from five different studies that detection and correction of a design error during the design phase required one-tenth of the effort to correct than if the error was found during the system testing phase, and one-hundreth if it was found during system operation and maintenance.

Other works have presented parametric models to represent the relationship between productivity and quality (*c.f.* Putnam *et al.* 1992, Ferdinand 1993). However, most of these studies have not gone without criticism. For example, Jones (1986) stated that the "cost to correct a defect" studies such as those cited by Boehm, were inaccurate because of the large fixed costs associated with defect removal efforts. The author further discussed other problems associated with such statistical studies, and warned of some misleading results. Also, as pointed out by several authors, the parametric models may impose much untested structure (*c.f.* Jeffery 1987, Banker *et al.* 1989) and may require extensive and continuous calibration for use in a particular organization (Kitchenham *et al.* 1984, Jeffery and Low 1990). Therefore, it remains a difficult task for management to get accurate estimates of the benefits of improving software quality, such as improved productivity or efficiency. These are vital for the justification and successful implementation of software quality improvement programs within an organization.

As is common through all business sectors and, in particular, service businesses, *performance ratios* (such as productivity and quality, as mentioned above) are widely used measures of software project success. Each of these ratios gives only a one dimensional, incomplete, picture of the project's "health". This results in management reports with a multitude of partial measures. Management and staff must then combine and interpret the information in a meaningful way to get an overall picture of performance. Because of its ability to handle the multidimensional nature of inputs and outputs in production processes, *Data Envelopment Analysis* (DEA) can be used effectively to resolve many of the problems inherent in the use of performance ratios. DEA is now well established as a powerful tool that supplements traditional approaches and provides further comprehensive insights into an organization's performance.

This paper presents two empirical DEA studies conducted at two large Canadian Banks on the efficiency of the software production and maintenance processes. Canada has six very large banks that constitute the core of the financial services industry. These banks have annual budgets for information technology in the range of several hundred million Canadian dollars, a significant portion of which is dedicated to software projects. Accordingly, bank management is especially interested in software project performance measurement, and its application to quality improvement and cost reduction. The current measurement programs at both banks involve the calculation of several performance ratios. It is our goal to show that the application of DEA, *with the data that is already being collected by the Bank*, leads to a clearer and more objective picture of performance. Furthermore, we show how DEA can be used as a general framework for resolving some particular issues of interest to management, and how its application may suggest further possible improvements to the measurement program. We further address two issues of great importance to industry practitioners: the explicit introduction of subjective managerial production goals and the vital role of quality when measuring efficiency.

For these studies, we introduce a new model of the software production process based on the compiled measures at both banks. The model reflects management's experience in both banks, and is consistent with the work published in the literature. We further point to difficulties in the modeling process and suggest possible improvements.

The first study analyzes a group of software development projects and compares the performance ratio approach, currently used at the bank, with the results obtained from DEA. In this study, we further use an extension of DEA with multiplier constraints for the identification and measurement of inefficiencies with respect to *subjective managerial goals*. This model leads to sharper efficiency estimates by incorporating (subjective) managerial information into the analysis; a procedure analogous to the estimation of overall efficiency, as opposed to pure technical efficiency (the product of standard DEA). By these means, we can distinguish, and quantify, the part of the inefficiency that arises from not fully exploiting production possibilities and that due to a lack of fulfillment of managerial goals. As is shown in this study, incorporating subjective value into the efficiency assessment must be done in a careful and rigorous manner, while understanding the framework that management is familiar with, in order to obtain meaningful results. We further point to different problems that usually arise when trying to capture managerial preferences for modelling purposes.

The second case study investigates the effect of quality on a group of software maintenance projects. Following a basic DEA analysis, we quantify the average effect of quality on the efficiency and cost of the projects. By subsequently using simpler production models, we further address some problems that result from the exclusion of quality from the models used to measure efficiency (and productivity), as is commonly the case.

This paper is further organized as follows. Section 2 gives a short overview of software production models in the current literature, and introduces the model used throughout this work. Sections 3 & 4 present the studies at the first and second bank, respectively. Section 5 gives some concluding remarks and recommendations. DEA models with constrained multipliers are briefly introduced in the appendix. For general presentations on DEA, the reader is referred to Norman and Stoker (1991), Fried *et al* (1993), and Charnes, Cooper, Lewin and Seiford (1995); for some comprehensive treatments of DEA models with multiplier constraints *c.f.* Charnes *et al.* (1990), Thompson *et al.* (1990), Ali and Seiford (1993).

2. Software Production Model

Cost estimation represents perhaps the majority of the effort expended to relate software production measures. Cost estimation models are usually the result of extensive analysis of large databases of projects where the main goal is to examine relationships between project cost (or effort) and various project factors, for predictive purposes. The reader is further referred to Boehm (1981), Conte *et al.* (1986), Kemerer (1987), and Fenton (1991), for some comprehensive reviews on cost estimation.

In contrast, within an efficiency measurement framework, one is more interested in assessing how well a group is using its resources to obtain a desired outcome; alternatively, one may want to assess how good an outcome one is producing with the given resources. Thus, one is intuitively interested in defining the *main resources* (inputs) and the *relevant products* (outputs) of the process, and in finding appropriate measures for these attributes.

Some applications of DEA to measure the efficiency of software production have been reported in the literature. The production models used in these studies are summarized in Table 1. All the models have Labour, measured either in labour hours or cost ($), as its main input representing the *effort*. The main output in the models is the *size* of code delivered. The most common measure for this attribute is the number of source lines of code (SLOC). However, in recent years, the number of *function points* (FPs) has gained acceptance as a more reliable measure of "size" (Albrecht 1979). The number of FPs may be directly computed from the software specifications and, thus, may also be helpful for cost predictions. In spite of its current popularity, the function point measure has not gone without some criticism (*c.f.* Fenton 1994, and the references cited therein).

Table 1: Review of Software Production models in DEA studies.

Study	Inputs		Outputs		
	Labour (hrs or $)	Other Expenses	Function Points	SLOC (lines of code)	Quality
Banker and Kemerer, 1989	x			x	
Elam, 1991	x	x	x		x
Banker, Datar and Kemerer, 1991	x		x	x	

The simple (one-input/one-output) production model by Banker and Kemerer (1989) was used a to estimate the *most productive scale size* of software development projects. Later, Banker *et al.* (1991) used both SLOC and FPs in order to study the effects of project characteristics on different phases of the software maintenance project life-cycle. The model used by Elam (1991) considered also a quality attribute of the software as output and was used for management applications. A limitation of this study was that most measures were normalized: labour cost per employee; FPs per work-month; and quality was measured by the total rework hours per FP. This normalization may be undesirable since it removes the scale component from the analysis.

In this study we propose a software production model with one input and three outputs as shown in Figure 1. The model reflects management experience at both banks where the studies were conducted. Hence, we only use the information deemed relevant, and already gathered by the banks. This allows us not only to incorporate fully the management objectives and knowledge of the process, but also enables us to show the benefits of DEA as compared to the current performance measurement techniques.

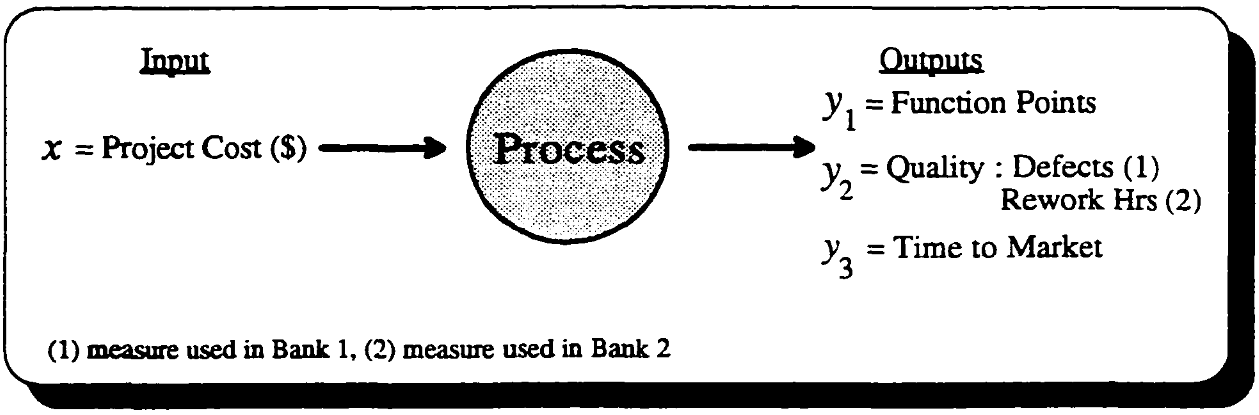


Figure 1. Software Production Model.

The single input, project cost, is a measure of effort and reflects the development cost of the project which includes labour, overhead, computer charges, etc. The costs are those charged to the client business unit served by the software production unit and this may differ somewhat from the actual cost incurred by the software production department (due to the accounting methods employed).

The first output is the size of the project. Management at both banks believed that the number of FPs was a useful measure for size and, thus, had ongoing FP measurement programs. For the second output, quality, each bank collected a different measure: 1) the number of defects detected in the four month period after implementation, and 2) the rework hours from final independent testing. Note that in contrast to the measure used in Elam 1991 (rework hrs/FP), both are non-ratio measures of quality. The rework hours measure considers, to a certain degree, the severity of the defect in terms of resources expended to correct it. However, it is much more difficult and time consuming to gather than a defect count and, thus, is less popular in industry.

Finally, the third output, time to market, measures the calendar duration of the project, and addresses its tradeoffs with the other outputs. In the study at the first Bank, this included analysis, design, coding and final independent testing. However, at the second Bank, this measure (consistent with rework and cost data) did not capture the calendar time of rework and independent testing.

Production models assume that the producer is interested in *maximizing its outputs* and/or *minimizing its inputs*; the fewer resources consumed and the larger the outcomes, the better. Since lower levels of the "quality" measures (defects or rework hours) and shorter times to market are preferred, one could argue that these variables should be treated as inputs. However, the main interest of our analysis lies in the cost savings achievable through efficiency improvements. Hence, it is convenient to use an input-oriented DEA model for efficiency evaluation with these two variables as outputs. This is also reasonable since, in general, the functionality of a particular project is specified and programmers do not really have direct control over the number of defects (one could argue that although they cause the defects, it is not done knowingly). Moreover, in most cases, the "market" imposes certain demands on the software producers regarding the project duration. This latter issue may vary depending on the application, customer base, etc. Therefore, we do not view quality and time to market as consumable production resources, but as important cost drivers. Using these two variables as outputs of the production process requires a transformation of the original raw data. Thus, the variables that enter the model, y_2 and y_3 , are obtained by subtracting the original measurement from the maximum observed output value in the sample. An advantage of this type of linear transformation is that the transformed variables preserve the units of the original data.

The production model in figure 1 describes better the production process than those of previous studies and incorporates many of the tradeoffs considered significant by management at both banks as well as by other software practitioners (*c.f.* Putnam and Meyers 1991). However, it is important to point out some limitations of the model and some possible improvements. First, the measures used for quality (number of software defects and rework hours) correspond only to a very narrow definition that misses many of the quality attributes suggested by McCall *et al.* (1977), Boehm *et al.* (1978), and others. The measures used may be reasonable from a developer's perspective. However, they cannot be seen as appropriate measures from the user's perspective since some empirical studies suggest little correlation between defects and failures during actual operation (Fenton 1994). Furthermore, other significant attributes related to quality, such as customer satisfaction, should be important additions to the model. Hence, serious effort is being undertaken at both corporations to include this in their measurement process.

The present model does not include *environmental factors* (such as hardware platform, tools, and staff experience) that may be relevant in the production process (*c.f.* Kemayel *et al.* 1991). Hence, the effect of these factors will be implicit in the efficiency measures. In future DEA analyses, these factors may be incorporated into the production model as exogenous (uncontrollable) inputs or outputs, or as categorical variables (*c.f.* Ali and Seiford 1993). They may also be used at a later stage as control variables for further hypothesis testing.

3. Study 1

We analyzed a group of eleven recently completed software development projects in the first Bank. The projects had been previously selected from a larger pool by removing those that were deemed to have a sizable portion of costs resulting from performance or software upgrades. Our main objectives were *i*) to compare the performance ratio approach to DEA and *ii*) to investigate different methods of incorporating managerial, subjective, information for constraining the DEA multipliers. This further resulted in classifications of the projects in terms of *effectiveness* (*i.e.* of some specific managerial goals). We contrasted these results to the qualitative classifications obtained by the Bank staff from performance ratios.

3.1 Analysis with Performance Ratios

Partial productivity ratios, and other performance ratios, have been widely used measures for determining the success of software projects. One of the reasons for the popularity of these ratios is their simplicity. In general, a single number on its own conveys little information about the performance of a project. These ratios allow for broader comparisons, by "normalizing" results for projects of different sizes. Although many measures are used to assess software project performance, the bank relies most heavily on three particular ratios:

- R_1 = Project Cost / FP (3.1)
- R_2 = Defects / 100FP
- R_3 = Project Duration / FP

Experts then use their judgement to combine the three ratios in order to classify the projects. The basic data and the three ratios of the eleven projects are given in table 2 together with one qualitative classification given by a Bank measurement expert.

Table 2: Data Set for Software Projects, Partial Ratio Measures *

Project	Cost	FP	Defects	Duration	R1	R2	R3	Classification
1	186	134	8	12	1,388.1	5.97	1.8	Worst
2	451	548	43	16	823	7.85	0.6	Intermediate
3	675	791	16	19	853.4	2.02	0.5	Best
4	338	375	16	10	901.3	4.27	0.5	Intermediate
5	315	358	12	15	879.9	3.35	0.8	Intermediate
6	152	144	1	4	1,055.6	0.69	0.6	Intermediate
7	338	373	13	12	906.2	3.49	0.6	Intermediate
8	386	269	52	15	1,434.9	19.33	1.1	Worst
9	278	329	7	12	845	2.13	0.7	Best
10	198	128	6	12	1,546.9	4.69	1.9	Worst
11	871	1,118	23	21	779.1	2.06	1.4	Best

* [Cost] = ($000's) , [Duration] = workmonths (wm; 1wm=20 calendar days),
[R_1] = $/FP , [$R_2$] = Defects/100FP , [$R_3$] = days/FP

Accordingly, projects 3, 9 and 11 were found to be relatively efficient ("best"). Projects 2, 4 and 7 were not considered efficient ("intermediate") primarily due to "high" defects/100FP ratios.

Project 6 was not considered efficient due to a "high" cost/FP ratio. Similarly, project 5 and 9 also considered as "intermediate" due to "high" delivery days/FP ratios. Projects 1, 8 and 10 were judged to be the most inefficient ("worst") due to relatively high values for all three ratios. It is important to point out that other Bank experts gave different classifications. This further demonstrates the subjective nature and some of the difficulties associated with interpreting such multiple-ratio measures.

At first glance, performance ratios seem easy to calculate and, hence, to use. However, while the process may be useful to obtain *qualitative* relative efficiency measures and/or classifications, the method is prone to several problems:

- Interpreting the partial information provided by each ratio is difficult and, in general, a highly subjective task. Using the analyst's judgement, the ratios are usually combined in order to obtain a "fuzzy" efficiency measure. This process can be seen as one where a weighting scheme is implicitly used but, on many occasions, not explicitly defined.
- Even when the weighting scheme is explicitly defined, several problems still exist. First, the choice of weights may be very subjective. Thus, the methodology further clusters, and confounds, *technical* and *allocative* inefficiencies, without being able to assess their respective impacts. Moreover, studies have shown certain instability of weight elicitation methods due to behavioural influences on weight judgement; for example, properties of the analysis such as attribute ranges, measurement scale, etc. have been shown to have strong effects on the judgement (*c.f.* Weber *et al.* 1993). As will be seen later in this section, this is a problem we face as well, although perhaps more subtly, when trying to incorporate managerial judgement in DEA.
- Direct comparison of individual ratios for different production units clearly does not account for possible scale effects or for tradeoffs from the different product attributes.
- In addition, the final qualitative classifications and measures of productive efficiency provide, at best, only a weak link between measurement and action.

3.2 Application of DEA.

Many of the deficiencies of the analysis using performance ratios can be overcome with DEA. As a first step, we obtained results for technical efficiency using the CCR and the BCC input-oriented models. The efficiency scores and reference sets (peer groups) for all the projects are given in table 3.

Table 3: DEA Efficiency scores, Peer groups

Project	CCR (constant RTS)		BCC (varible RTS)	
	θ (efficiency)	Peer group	θ (efficiency)	Peer group
1	0.74	$\lambda_6 = 0.824$ $\lambda_9 = 0.0466$	0.82	λ_6=1.0
2	0.97	$\lambda_6 = 0.308$ $\lambda_{11} = 0.4505$	0.98	$\lambda_9 = 0.722$, $\lambda_{11} = 0.278$
3	0.93	$\lambda_9 = 0.424$ $\lambda_{11} = 0.583$	0.93	$\lambda_6 = 0.164$, $\lambda_9 = 0.212$ $\lambda_{11} = 0.624$
4	0.94	$\lambda_6 = 0.653$ $\lambda_{11} = 0.251$	0.94	$\lambda_6 = 0.429$, $\lambda_9 = 0.413$ $\lambda_{11} = 0.159$
5	0.94	$\lambda_9 = 0.845$ $\lambda_{11} = 0.0717$	0.95	$\lambda_9 = 0.963$, $\lambda_{11} = 0.037$
6	1	λ_6 =1.0	1	$\lambda_6 = 1.0$
7	0.92	$\lambda_6 = 0.346$ $\lambda_9 = 0.1835$	0.92	$\lambda_6 = 0.085$, $\lambda_9 = 0.839$ $\lambda_{11} = 0.076$
8	0.58	$\lambda_6 = 0.378$ $\lambda_{11} = 0.192$	0.61	$\lambda_6 = 0.324$, $\lambda_{11} = 0.676$
9	1	$\lambda_9 = 1.0$	1	$\lambda_9 = 1.0$
10	0.69	$\lambda_6 = 0.904$	0.77	$\lambda_6 = 1.0$
11	1	$\lambda_{11} = 1.0$	1	$\lambda_{11} = 1.0$

As can be seen from the table, the results are quite similar with both DEA models. In particular, three projects (DMUs 6, 9 and 11) were found to be both technical and scale efficient (θ=1 and zero slacks in the BCC and CCR respectively). Thus, there does not seem to be any indication of scale effects within the ranges of the sample. Also notice that only three groups (1,8 and 10) have efficiency scores below 90%.

Upon examination of the peer groups, management believed that the results were not practically meaningful for them. In particular, of the 8 inefficient units, 5 have DMU_{11} on their CCR reference set (6 of them in BCC). Many of these DMUs were substantially smaller in size (number of FPs) than DMU_{11} . Bank management had previously categorized projects according to size; projects of over 1000 FPs were in a *medium* size category, and those of under 1000FP were included in the *small* category. They believed that comparing small projects with medium size projects might not be fair. Furthermore, it was important for management to arrive at targets that were consistent with their experience, and setting efficient targets from a combination of projects in different categories was not meaningful since the project characteristics are quite different. Thus, the results in table 3 do not necessarily indicate possible transferable best practices for the inefficient units.

Similar problems to this have long been identified in the DEA literature and can be readily solved introducing a categorical variable into the DEA model (*c.f.* Banker and 1986a, Ali and Seiford 1993). In this case, the introduction of the categorical variable simply implies rerunning the DEA model for DMUs 1 to 10 (small size) without DMU_{11} in the sample; the results did not change for DMU_{11}.

The new efficiency scores and peer groups are presented in Table 4. Note that, in general, the efficiency scores did not change dramatically. However, DMU_2 with a previous score of 0.97, not only became efficient but also appears in the reference set of four of the seven remaining inefficient units (in CCR).

Table 4: DEA Efficiency scores, Peer groups: Results with Categorical Constraints

Project	CCR (constant RTS)		BCC (varible RTS)	
	θ (efficiency)	Peer group	θ (efficiency)	Peer group
1	0.74	$\lambda_6 = 0.824$ $\lambda_9 = 0.0466$	0.82	$\lambda_6 = 1.0$
2	1	$\lambda_2 = 1.0$	1	$\lambda_2 = 1.0$
3	0.97	$\lambda_2 = 1.105$ $\lambda_9 = 0.564$	1	$\lambda_3 = 1.0$
4	0.94	$\lambda_6 = 0.0442$ $\lambda_9 = 1.121$	0.98	$\lambda_2 = 0.245$, $\lambda_3 = 0.204$ $\lambda_6 = 0.551$
5	0.96	$\lambda_2 = 0.136$ $\lambda_9 = 0.862$	0.96	$\lambda_2 = 0.132$, $\lambda_9 = 0.868$
6	1	$\lambda_6 = 1.0$	1	$\lambda_6 = 1.0$
7	0.93	$\lambda_2 = 0.125$ $\lambda_9 = 0.925$	0.94	$\lambda_2 = 0.173$, $\lambda_3 = 0.0735$ $\lambda_6 = 0.151$, $\lambda_9 = 0.602$
8	0.59	$\lambda_2 = 0.110$ $\lambda_9 = 0.634$	0.61	$\lambda_6 = 0.324$, $\lambda_{11} = 0.676$
9	1	$\lambda_9 = 1.0$	1	$\lambda_9 = 1.0$
10	0.69	$\lambda_6 = 0.904$	0.77	$\lambda_6 = 1.0$
11	1	$\lambda_{11} = 1.0$	1	$\lambda_{11} = 1.0$

3.3 Overall Efficiency: DEA Models with Multiplier Constraints.

Management considered it important to obtain efficiency measures that account for internal policies of the Bank's IS department. This is analogous to investigating overall (and, hence, also allocative) efficiency. However, in this case, the traditional (economic) definition of allocative efficiency in terms of market prices is broadened to account for "prices" representing other behavioural goals; in particular, the satisfaction of internal (subjective) management policy regarding the mix of software projects' cost, quality, duration and size. It is important to note further that these policy preferences might not coincide with those of the customers (*i.e.* with the preferences of the software users).

For this purpose, in the second stage of the analysis, we turned to managerial information in order to constrain further the DEA multipliers and, hence, to tighten efficiency estimates. DEA models with multiplier constraints are given in the appendix; for some comprehensive treatments of these models, the reader is further referred to Charnes *et al.* (1990), Thompson *et al.* (1990), Ali and Seiford (1993). We investigated two different methods for eliciting the weight constraints from management's subjective information.

3.3.1 Z-Score Approach for Constraining Weights

The first method explicitly considers management's own framework for analyzing software project performance. Traditionally, management viewed efficient projects as those that achieved a proper mix of the three ratios (3.1): R_1=cost/FP , R_2=Defects/100FPs , R_3=days/FP . Obviously, there is no unanimous consensus of what constitutes a "proper mix".

We gathered the (subjective) opinions of six experienced managers on the "percentage importance of each of these ratios in an efficiency evaluation". This indicated how much a particular ratio, in the eyes of a manager, should contribute to an overall efficiency score. We used this information to bound the DEA multipliers as explained below.

The suggested performance evaluation based on the ratios can be understood in terms of a z-score, familiar to the economic literature. In this approach, the efficiency of a particular project can be constructed as a weighted sum of several ratios. The z-score is maximum for efficient units and we can express the z-score of a project *k* as:

$$Z_k = w_1 z_{1k} + w_2 z_{2k} + w_3 z_{3k} \quad , \quad k = 1, \ldots, n \tag{3.2}$$

In expression (3.2),

$$z_{1k} = y_{1k} / x_k , \qquad (3.3)$$

$$z_{2k} = y_{2k} / y_{1k} ,$$

$$z_{3k} = y_{3k} / y_{1k} ,$$

x_k denotes the single input and y_{jk} the j-th output (in the transformed DEA format), for project k; w_i are the (subjective) weights.

Note that while efficient projects tend to minimize ratios used by the bank (3.1), it is desirable that they maximize the ratios used in the z-score (3.2). Hence, the ratios (3.3) correspond to transformations of the ratios (3.1) in terms of the data that enters the DEA model and so that they meet the maximizing requirement of the z-score. Accordingly, z_1 is the inverse of R_1 since higher "productivity", measured as FP/$, is desirable. No inversion is necessary for the remaining ratios, z_2 and z_3, because of the data transformations described in section 2.

Then, for the k-th project, the percentage importance of the j-th ratio is given by $w_j z_{jk} / \Sigma_i w_i z_{ik} \equiv w_j z_{jk} / Z_k$. Therefore, the information gathered from management can be used to obtain bounds of the form:

$$a_j \leq \frac{w_j z_{jk}}{Z_k} \leq b_j \quad , \quad j = 1, 2, 3 \qquad (3.4)$$

with $a_j \leq b_j$.

In this study, we used as the interval of importance for each ratio, $[a_j, b_j]$, the union of the intervals gathered from each of the six managers. In particular, the resulting upper and lower bound vectors, **a** and **b**, obtained for this study are as follows:

$$\mathbf{a} = \begin{pmatrix} 15 \\ 25 \\ 10 \end{pmatrix} , \quad \mathbf{b} = \begin{pmatrix} 60 \\ 60 \\ 50 \end{pmatrix} \qquad (3.5)$$

This implies, for example, that when evaluating the efficiency of each project k, ratio z_1 (FP/$) should contribute between 15% and 60% to the total score. Intuitively, the bounds (3.5) reflect the fact that, overall, managers thought that good projects should have performed (similarly) well in all the ratios.

The set of constraints (3.4) can be used now to obtain the corresponding multiplier bounds for the DEA program. First we eliminate the presence of Z_k by dividing any two constraints. Clearly, this leads to:

$$\frac{a_i}{b_j} \leq \frac{w_i z_{ik}}{w_j z_{jk}} \leq \frac{b_i}{a_j} \quad , \quad i,j = 1,2,3 \quad \text{and} \quad i<j \qquad (3.6)$$

It is a simple exercise in algebra to show further that expressions (3.3) and (3.6) can be used, in this case, to bound the DEA output multipliers in the following way:

$$\frac{a_i\, y_{jk}}{b_j\, y_{ik}} \leq \frac{\mu_i}{\mu_j} \leq \frac{b_i\, y_{jk}}{a_j\, y_{ik}} \; , \quad i,j = 1,2,3 \quad \text{and} \quad i<j \, ; k = 1,\ldots,n \qquad (3.7)$$

where μ_i represents the i-th output multiplier and, again, y_{ik} denotes the i-th output for DMU_k. These constraints are similar those given in (Wong & Beasley 1990). However, they are expressed in a somewhat simpler and more useful form.

For each DMU, the set of constraints in (3.7) leads to 6 linear homogeneous inequalities that represent a convex cone in the output multiplier space. Note that they are dependent on the particular unit k and, hence, are very sensitive to the observed data, as are those in (Wong & Beasley 1990). Thus, there are different ways for implementing these types of constraints in a DEA analysis. In particular, we implemented three different approaches to analyze the 10 DMUs in the "small" category (DMUs 1 to 10).

a) In the first set, we performed several DEA analyses with the constraints (3.7) obtained for each DMU in the sample; *i.e.* we performed 10 different analyses using in each of these a different DMU as a basis to obtain the weight constraints (3.7). Hence, for each of the k=1,...,n analyses, we append the following constraints to the basic DEA model:

$$\frac{15\,y_{2k}}{60\,y_{1k}} \le \frac{\mu_1}{\mu_2} \le \frac{60\,y_{2k}}{25\,y_{1k}}, \qquad \frac{15\,y_{3k}}{50\,y_{1k}} \le \frac{\mu_1}{\mu_3} \le \frac{60\,y_{3k}}{10\,y_{1k}}, \qquad \frac{25\,y_{3k}}{50\,y_{2k}} \le \frac{\mu_2}{\mu_3} \le \frac{60\,y_{3k}}{10\,y_{2k}}, \quad k = 1,\ldots n. \quad (3.8)$$

The first five columns of table 5 present some of the results using the CCR model for this case. We used primarily the CCR frontier since no significant difference was observed between the results with the CCR and BCC models. Notice that, in all cases, DMU_6 was the only efficient project. Results were also obtained using the BCC model. For this model, only in a few cases did DMU_2 and DMU_9 obtain an efficiency score of unity, in addition to DMU_6. Perhaps the most interesting consequence of this analysis arises from the fact that DMU_6 was not deemed particularly efficient by management in their subjective evaluation of the ratios (see table 2), since it had a high [$/FP] ratio, as was mentioned earlier in section 3.1 . The present analysis led management to seriously reconsider their views.

[illegible] Efficiency Scores with different sets of Constraints (CCR model).

DMU	DMU used in expression (3.7) to obtain the constraints						Bounds on Acceptable Trade-offs
	DMU_1	DMU_2	DMU_3	DMU_6	DMU_9	Average DMU	
1	0.7	0.69	0.61	0.71	0.68	0.68 (2)	0.71 (2)
2	0.78	0.18	0.22	0.84	0.52	0.45 (9)	0.25 (9)
3	0.76	0.24	0.2	0.83	0.52	0.46 (8)	0.30 (8)
4	0.81	0.39	0.4	0.86	0.62	0.58 (6)	0.44 (7)
5	0.82	0.42	0.36	0.88	0.63	0.59 (5)	0.47 (5)
6	1	1	1	1	1	1 (1)	1 (1)
7	0.8	0.4	0.38	0.85	0.62	0.58 (7)	0.45 (6)
8	0.45	0.12	0.16	0.48	0.31	0.27 (10)	0.15 (10)
9	0.9	0.52	0.47	0.95	0.72	0.68 (3)	0.56 (4)
10	0.66	0.67	0.59	0.67	0.65	0.65 (4)	0.69 (3)

Two drawbacks are evident in this type of analysis. Not only is it computationally intensive, but it also leads to large tables of numbers that are difficult to interpret. Hence, it may only be useful for cases like this one, with a small number of DMUs and where some conclusions seem evident.

b) One straightforward way to obtain an "average" of the previous results is to run only one DEA program (for all DMUs) with the constraints (3.7) given for an *average project.* This requires to run the DEA program only once. The results for this case are given in the sixth column of Table 5. The number in parenthesis gives the ranking of the projects based on their efficiency scores.

c) The tightest constraints are obtained by requiring that (3.7) be satisfied simultaneously for *all* DMUs. As implemented by Wong & Beasley, this leads to $2\times(s\times n)$ constraints to be appended to the basic DEA model (66, in this case). However, by expressing them in ratio form, as in (3.7), it is straight forward to show that this approach leads to the following equivalent, yet simpler, set:

$$\max_k\left[\frac{a_i\, y_{jk}}{b_j\, y_{ik}}\right] \le \frac{\mu_i}{\mu_j} \le \min_k\left[\frac{b_i\, y_{jk}}{a_j\, y_{ik}}\right] \quad , \quad i,j=1,2,3 \quad \text{and} \quad i<j \tag{3.9}$$

Notice that the number of constraints has now been reduced to $2\times\binom{s}{2}$ (6, in this case). Moreover, by using expression (3.9) one may be able to infer whether the solution to the problem is infeasible. This is not at all uncommon for these type of constraints, especially for wide ranges of data. For our particular problem, it was clear that the constraints led to infeasible solutions, since the lower bound was larger than the upper bound in some ratios.

3.3.2 Using Information on Tradeoffs to Obtain Weight Constraints

The second method used for weight bound elicitation is based on management's judgement of realistic (and desired) ranges for *tradeoffs* of different variables. On a DEA frontier, tradeoffs between two variables are simply given by the ratios of multipliers (*c.f.* Charnes *et al.* 1985). This is perhaps a preferred way from a DEA standpoint since it leads directly to upper and lower bounds on these ratios.

In this study, managers believed that they could give only "acceptable" upper bounds for ratios of variables they were used to handling in their previous performance evaluations (ratios (3.1)). However, they considered this information adequate for this purpose. The resulting constraints on the ratios of multipliers, to be appended to the basic DEA formulation, are:

$$\frac{\mu_1}{v} \leq 1.5 \ , \quad \frac{\mu_1}{\mu_2} \leq 0.08 \ , \quad \frac{\mu_1}{\mu_3} \leq 0.1 \qquad (3.10)$$

where v denotes the single input multiplier and μ_i the output multipliers. Expression (3.10) leads to three homogeneous linear inequality constraints, describing a linked input-output cone.

The efficiency scores for this case are given in the last column of table 5. Also in parenthesis is the ranking of each project, relative to these "acceptable" bounds. The salient, and perhaps surprising, aspect of the analysis is that, in spite of the fact that the bounds in expression (3.10) seem loose, and of the difference in elicitation techniques, the results compare quite well with those obtained in the *z*-score approach. The fact that DMU_6 was once again the only efficient project stands in sharp contrast to the results of the performance ratio approach (see table 2). Furthermore, although the scores were in general a little lower in this case than in the z-score approach, the ranking of projects stayed essentially the same.

3.3.3 Comparison of DEA Results to Performance Ratio Classifications

Both the DEA and performance ratio analyses were conducted using the same data and according to management's subjective goals. It should be further stressed that the managerial judgement used within the DEA analysis was elicited using the same performance ratio framework employed by management. As was mentioned earlier, the two different approaches to eliciting DEA multiplier restrictions gave almost identical rankings. However, some alarming differences were found when we compared these results to the classifications using performance ratios (see the last column of table 2).

For example, projects 3 and 9 had been classified as "best" performers by the bank expert using the performance ratio approach. Although DEA ranked project 9 quite high (third or fourth), there is clearly a major discrepancy found in the performance of project 3 (ranked third last). The bank expert had further classified projects 1, 8 and 10 "worst" performers, using the performance ratio approach. While project 8 also ranked poorly with DEA (ranked last), project 1 and project 10 were very good performers (second and third or fourth, respectively). Furthermore, as mentioned earlier, project six consistently and strongly dominated the DEA analysis; yet, it had been classified only as an "intermediate" performing project.

The discrepancy between an individual's subjective classification and his or her explicitly stated goals can be best explained by the multi-dimensionality of the production data and the simultaneous consideration of judgement regarding multiple producer objectives. This results in too daunting and complex a task for an analyst using multiple performance ratios. It is our conviction that DEA provides a more consistent and systematic method of incorporating judgement into performance analysis. Moreover, it leads to a clearer and more objective picture of overall performance.

3.3.4 Remarks

It is important to stress that extreme caution is necessary in defining the weight-constraint elicitation techniques, and in posing the relevant questions in order to obtain the key information. As has been cautioned also in other related areas of Operations Research, practitioners should be aware of the many difficulties and inconsistencies they may encounter when trying to capture managerial information for more rigorous modelling and analysis. In particular, some problems that arise from the behavioural influences on weight judgement have been addressed in the Multi-Criteria Decision Analysis literature (*c.f.* Weber *et al.* 1993). However, this is a problem that has not been explicitly discussed throughout the DEA literature. It is also very important to realize that the (subjective) information must be interpreted within the framework familiar to management. Not only does this point to more reliable information obtainable from them, but also to the limitations of the results and the biases that must be overcome for more rigorous and meaningful analyses. An effective implementation of these techniques requires the continuous interaction of the analyst with management at several stages. The results are fed back to management at each stage, and discussed with the analyst. This should ultimately result in consistent bounds for the multipliers. Moreover, it should lead to a more thorough understanding of the production process, and a more explicit policy relevant to the set of revised managerial goals.

4. Study 2

The study at the second Bank involved a sample of fifteen completed software maintenance (enhancement) projects. Here, quality was measured as rework hours used to repair defects, as opposed to simply the number of defects. Furthermore, project costs did not include final independent testing, and the resulting development rework.

At this bank, management had established a slightly more developed measurement program with a strong focus on the quality of software produced. They firmly believed that improved software quality would greatly benefit both the bank and its customers. Thus, we used DEA in this study as a framework to investigate the effect of quality on the cost and efficiency of software production. Furthermore, we investigated problems that can arise due to the exclusion of quality measures from the production model. This is the case when the focus of performance is placed primarily on "productivity", defined in a simplistic way as the ratio of size to effort expended.

We begin this analysis with the application of input-oriented CCR and BCC models. The raw data and efficiency scores are given in table 6. Notice that five projects are found efficient with the BCC model while only two with the CCR. Further investigation of returns to scale intervals indicate that DMUs 6, 8 and 15 were operating under decreasing returns to scale. This suggests that the process may not be well described by a constant returns to scale technology. Hence, we further used the results obtained from a variable returns to scale technology to investigate the impact of quality on project cost and efficiency.

Table [illegible]: Study 2 Data Set (Untransformed) and Efficiency Scores.

Project	Cost ($000's)	FP	Rework (hours)	Duration (days)	θ (efficiency)	
					CCR (constant RTS)	BCC (variable RTS)
1	1,162	557	940	420	0.42	0.72
2	766	485	543	420	0.56	0.86
3	378	108	319	420	0.44	0.51
4	421	165	316	209	0.61	0.62
5	194	169	58	324	1	1
6	961	609	220	381	0.56	1
7	673	183	451	235	0.34	0.37
8	1,279	509	344	203	0.35	1
9	284	119	272	230	0.87	0.88
10	507	74	313	180	0.55	0.63
11	294	334	387	153	1	1
12	348	118	617	262	0.58	0.66
13	598	194	443	305	0.33	0.35
14	759	462	1,871	318	0.54	0.8
15	471	32	262	165	0.62	1

4.1 The Effect of Quality on Efficiency and Project Cost

The bank categorizes projects in terms of quality according to "percentage rework": the ratio of rework hours (resulting from independent testing) divided by total development hours. Projects with a rework ratio greater than 10% are classified as *unacceptable*, otherwise they are classified as *acceptable*. These quality categorizations were used to address the following research question:

- What is the average difference in efficiency and cost between projects with *unacceptable* and *acceptable* quality?

For this purpose, we divided the projects into two sets according to their quality categories, in order to compare the mean efficiencies of each set. The differences in these mean efficiency scores are used to estimate the impact of quality. Clearly, hypothesis tests (*c.f.* Banker *et al.* 1990) regarding the impact of quality on the efficiency scores are not appropriate since quality is a factor already included in the DEA model and, thus, directly affects the efficiency scores. Moreover, the sample size was too small to obtain meaningful statistics. Table 7 lists the average efficiencies for each set of projects, and for the whole sample, along with the average rework percentages (**aq**, **uq** denote the set of *acceptable* and *unacceptable* quality projects respectively, and **avq** denotes the entire set of projects).

Table 7: Average Efficiency of Different Project Quality Categories/Sets

	Acceptable Quality **(aq)**	**Average Quality** **(avq)**	**Unacceptable Quality** **(uq)**
Average Efficiency	0.87	0.76	0.69
Average Rework %	6.7	11.8	15.2

As can be seen from the table, there is an 18% difference in the average efficiency of *acceptable* quality projects and *unacceptable* quality projects. However, due to the small sample size, these averages are quite sensitive to changes in individual DMU efficiency scores as well as to changes to the 10% rework categorization criterion. For instance, if the rework criterion is changed from 10% to 10.5%, the difference in efficiency between the two quality categories drops to 11%.

The investigation of the effect of improving quality on project cost can be conducted in a similar manner. For this purpose, let us introduce a cost function that describes the relationship between the project cost, the efficiency scores θ, and the output vector. By noting that the single input used in the model is cost, the DEA frontier may be used to represent a minimum cost function for all feasible output vectors. Following Banker *et al.* (1990), this cost function can be expressed as:

$$C=\psi\times f(\mathbf{y}) \tag{4.1}$$

where $\mathbf{y}$ is a vector of outputs and $\psi = 1/\theta$. Averaging (4.1) over the entire observed sample leads to:

$$\bar{C}_{avq}=\bar{\psi}_{avq}\times\overline{f(\mathbf{y})} \tag{4.2}$$

where ($\overline{}$) is used to denote averages and the subscript *avq* denotes averages of the entire sample. Then, we can calculate the average in cost difference between quality categories, $\Delta\bar{C}$, in the following way:

$$\begin{aligned}\Delta\bar{C} &= \overline{C_{uq}-C_{aq}} = \bar{C}_{uq}-\bar{C}_{aq}\\ &= (\bar{\psi}_{uq}-\bar{\psi}_{aq})\times\overline{f(\mathbf{y})}\\ &= (\bar{\psi}_{uq}-\bar{\psi}_{aq})\times\frac{\bar{C}_{avq}}{\bar{\psi}_{avq}}.\end{aligned} \tag{4.3}$$

where the subscripts *aq*, *uq* denote that the average is over set of *acceptable* (**aq**) and *unacceptable* (**uq**) quality projects. Notice that we can perform the last step in (4.3) since both sets of projects are evaluated against the same production frontier. Then, by simple substitution in (4.3) we obtain the difference in average project cost between the acceptable quality projects and unacceptable quality projects as:

$$\Delta\bar{C}=(1.577-1.337)\times\frac{\$606,000}{1.481}=\$98,200.$$

Furthermore, this cost difference is *in addition* to the 8.5% avoided rework.

It is important to note that this simple methodology to quantify the effects of quality is somewhat circular, given that the efficiency measures *already* contain a quality output. However, it does provide management an idea of the cost reductions and possible improvements that may be achievable through the introduction of quality improvement programs. These can be then weighed against the costs of introducing such policies. If the efficiency measures did not consider quality, they would be much less meaningful. We explore this case in the following section.

4.2 Efficiency Without Quality

In order to investigate further the impact of quality on efficiency measurement, let us exclude the quality measure (rework hours) from the production model. This exercise is useful in helping management understand the implications of efficiency measures that neglect quality, as is often the case with productivity ratios. For example, the average efficiency scores for each set (as previously categorized), obtained from the 1 input and 2 output model that results from excluding quality, are given in table 8. The numbers in parenthesis give the average efficiency when a 10.5% rework criterion is used instead of 10%. As can be seen, there is little difference in the average efficiency of the sets. Furthermore, if the 10.5 criterion is used, the set with *unacceptable* quality shows a higher average efficiency.

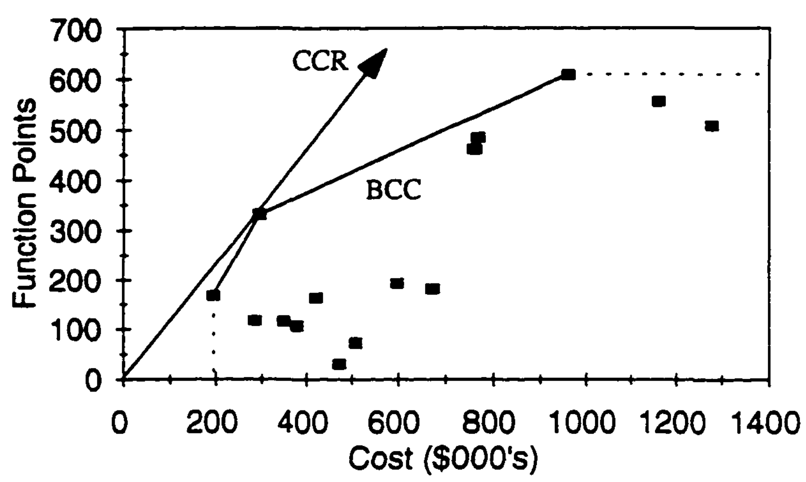
Table 8: Efficiency Without Quality.

	Acceptable Quality (AQ)	Unacceptable Quality (UQ)
Average BCC Efficiency Score (1 Input - 2 Output Model)	0.8 (0.76)	0.79 (0.82)
Average BCC Efficiency Score (1 Input - 1 Ouput Model)	0.7 (0.67)	0.71 (0.74)
Average CCR Efficiency Score (1 Input - 1 Ouput Model)	0.4	0.51

The second line in table 8 gives the results for a single input (cost) and single output (FP) model. Again, there is a rather small difference between the average efficiency of the *unacceptable* quality projects and the *acceptable* quality projects. Furthermore, the *unacceptable* quality projects actually have a *lower* average $/FP.

Notice that the 1 input and 1 output model leads to the Bank's definition of productivity. The results suggest, however, that this measure of productivity is at best unrelated but most probably inversely related to quality; such a measure may actually tend to "reward" bad quality. Furthermore, as with all ratio measures, it does not account for possible scale effects. As pointed out earlier, this seems to be important for this particular set of data. This is further shown in figure 2, where the number of FPs is plotted against cost and the BCC and CCR frontiers are both shown. The productivity measures ($/FP) correspond to CCR efficiency scores (once normalized by the highest $/FP ratio). Notice however, that the data seems to suggest significant scale effects on the relationship between these two variables.

Figure 2: Cost vs. FP



These findings are contrary to much of the conventional wisdom and indicate the importance of including quality in efficiency and productivity measures. Indeed quality should be included due to the tradeoffs that it has with the input as well as the other outputs, and especially since a great deal of effort is now expended on ensuring quality software. In this study we only incorporated a scalar measure, but future efforts will focus on including more attributes of quality in the software production model.

5. Concluding Remarks

This paper introduces a new model of software production that embodies more outputs, and hence more tradeoffs, than those previously cited in the literature. The production model builds on previously published work, and includes two new outputs: a non-ratio quality measure and a measure of project duration. It is important to stress that this model was developed in conjunction with bank management at two different institutions and, thus, it further reflects their views on the significant factors in the production process. It is our belief that working in association with management is a critical step in ensuring the acceptance and relevance of *any* DEA study or program. Therefore, we have conducted the analysis with the *same* data currently used by management to assess performance. This provides a good starting point for demonstrating the strengths of DEA in comparison to existing methodologies and results. Over time, the production model can (and, most likely, will) be modified to reflect better both the knowledge of the process and the (perhaps also clearer) managerial goals.

In the first study, the incorporation of subjective judgement into the efficiency measures was of considerable importance to management. In general, the justification for incorporating this judgement into DEA analyses may be twofold: to "correct" unrealistic regions of the technically efficient frontier, and/or to incorporate management goals and value judgements in order to move towards overall efficiency estimates. The latter was the focus in this study. In particular, we investigated two different methods for eliciting multiplier constraints from available managerial information. In spite of the difference in the elicitation techniques, the results of both methods compared quite well and the ranking of projects was almost identical. It is particularly important to stress that the managerial judgement gathered for this DEA study was elicited using the same performance ratio framework employed by bank management. By contrasting these results to those of the performance ratio approach, we demonstrate that, for this case, DEA provides a more consistent and objective view of performance.

It is very important to stress that the elicited information must be interpreted within the framework familiar to management. Surely, this results in more reliable information for the analysis. However, it is essential for a more rigorous and meaningful analysis, that the analyst be aware of the biases of such framework and, hence, of the limitations of the results. Thus, an effective implementation of these techniques should probably be evolutionary and requires the continuous interaction and feedback between the analyst and the managers. This should result in more consistent and meaningful multiplier bounds, based on a more refined set of goals, and in a greater understanding of the production process.

At the bank in the second study, a strong focus of the measurement program was the quality of software produced. Management was particularly interested in gauging the benefits of improved quality. Our results suggest that quality (as measured) has a sizable impact on project cost and efficiency. This is important in order to justify the implementation of software quality assurance programs within an organization. We have also shown the importance of including measures of quality in the production model, and highlighted problems that can arise from neglecting to do so. However, as mentioned earlier, the definition of quality used in this paper (and in most software performance analyses) is quite narrow and omits other important attributes, such as customer/user satisfaction. We expect the impact of a broadened definition of quality to be even more substantial. Thus, we believe that it is imperative for organizations to focus increasing efforts onto the measurement of quality attributes, and to include them in efficiency and productivity measurement programs.

Appendix: Basic DEA Models

In study 1, we incorporate managerial information to further constrain the DEA multipliers and obtain sharper efficiency estimates that accord to managerial goals. For completeness, in this appendix we briefly review DEA models with additional multiplier restrictions used in this paper.

Consider a set of n DMUs to be evaluated. Each DMU consumes different amounts of m inputs to produce s outputs. We denote by X, the $m \times n$ matrix of inputs with entries $x_{ij} \geq 0$ given by the amount of input i consumed by DMU$_j$; and by Y, the $s \times n$ output matrix, with entries $y_{rj} \geq 0$ given by the amount of output r consumed by DMU$_j$. Furthermore, we assume that each DMU has at least one positive input and one positive output.

One can incorporate additional managerial information in DEA by adding further linear homogeneous inequality constraints to the basic DEA multiplier program. Then, with a CCR model for example, the efficiency of a particular DMU, call it DMU$_0$, is given by:

$$\begin{aligned} \max_{u,v} \ & z = \mathbf{y}_o^T \mu \\ \text{s.t.} \ & \mathbf{x}_o^T \mathbf{v} = 1 \\ & \mathbf{Y}^T \mu - \mathbf{X}^T \mathbf{v} \leq \mathbf{0} \\ & \mathbf{A}^o \mu + \mathbf{A}^i \mathbf{v} \leq \mathbf{0} \quad , \quad \mu, \mathbf{v} \geq \mathbf{0} \end{aligned} \tag{A.1}$$

where $\mathbf{x}_o = (x_{1o}, x_{2o}, \ldots, x_{mo})$ and $\mathbf{y}_o = (y_{1o}, y_{2o}, \ldots, y_{so})$ are the input and output vectors of DMU$_0$ and $\mathbf{v} = (v_1, v_2, \ldots, v_m)^T$ and $\mu = (\mu_1, \mu_2, \ldots, \mu_s)^T$ are the vectors of input and output multipliers; $\mathbf{A}^i$ and $\mathbf{A}^o$ are $(k \times m)$ and $(k \times s)$ matrices, respectively; and k is the total number of multiplier constraints (to simplify the exposition, we have considered a model without non-archimidean infinitesimals). Note that the k extra constraints in (A.1) correspond to a new variable in the dual envelopment program.

This approach generalizes DEA models by requiring that input and output multipliers be restricted further to a given closed cone. When the multiplier constraints in (A.1) involve only input multipliers (*i.e.* $\mathbf{A}^o = \mathbf{0}$), it is called an input cone; if they relate only output multipliers ($\mathbf{A}^i = \mathbf{0}$), then it is called an output cone. The ratios may also correspond to separable input and output cones, when part of the constraints correspond exclusively to an input cone and the rest to an output cone; hence, the set of extra constraints can be written as:

$$\begin{bmatrix} \mathbf{A}^{o}_{\mu} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{i}_{\nu} \end{bmatrix} \begin{bmatrix} \mu \\ \upsilon \end{bmatrix} \leq \mathbf{0} \,.$$

When any of the constraints involves at least one input and one output, it is referred to as a linked cone.

A particular case of the multiplier constraints in (A.1) occurs when market prices or other managerial information is used to set bounds on ratios of pairs of multipliers; *i.e.*, if we denote the vector of multipliers by $\chi^{\mathrm{T}} = (\chi_1, \ldots, \chi_{m+s}) = (\mu^{\mathrm{T}}, \mathrm{v}^{\mathrm{T}})$, the constraints are of the form:

$$a_{ij} \leq \chi_i \,/\, \chi_j \leq b_{ij} \,,\; i \neq j \qquad \text{(A.2)}$$

with $0 \leq a_{ij} \leq b_{ij}$. These are sometimes referred to as *cone ratios*.

References

Abdel-Hamid, T. and S.E. Madnick (1991), *Software Project Dynamics An Integrated Approach*, Englewood Cliffs, NJ: Prentice Hall.

Albrecht, A.J. (1979), "Measuring Application Development Productivity," *IBM Applic. Dev. Joint Share/Guide Symp.*, Monterey, CA.

Ali, A.I. and L.M. Seiford (1993), The Mathematical Programming Approach to Efficiency Analysis, in *The Measurement of Productive Efficiency*, Fried H.O., Lovell K.C.A, Schmidt S.S. Editors, New York: Oxford University Press.

Banker, R.D., S.M. Datar and C.F. Kemerer (1991), "A Model To Evaluate Variables Impacting The Productivity Of Software Maintenance Projects," *Management Science*, 37, 1.

Banker, R.D., R.J. Kauffman and R.C. Morey (1990), "Measuring Gains in Operational Efficiency from Information Technology: A Study of the Positran Deployment at Hardee's Inc.", *Journal of Management Information Systems*, 7, 2.

Banker, R.D. and C.F. Kemerer (1989), "Scale Economies in New Software Development," *IEEE Transactions on Software Engineering*, 15, 10.

Banker, R.D. and R. Morey (1986), "The Use of Categorical Variables in Data Envelopment Analysis," *Management Science*, 32, 12.

Boehm B.W. (1981), *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall.

Boehm, B.W., J.R. Brown, J.R. Kaspar, M. Lipow, G.J. MacCleod and M.J. Merrit (1978), "Characteristics of Software Quality", North Holland.

Charnes, A., W.W. Cooper, B. Golany, L.M. Seiford and J. Stutz (1985), "Foundations of DEA for Pareto-Koopmans Efficient Empirical Production Functions," *Journal of Econometrics*, 30, 12.

Charnes, A., W.W. Cooper, Z.M. Huang and D.B. Sun (1990), "Polyhedral Cone-Ratio DEA Models With an Illustrative Application to Large Commercial Banks," *Journal of Econometrics*, 46.

Charnes, A., W.W. Cooper, A. Lewin and L.M. Seiford (1995), *Data Envelopment Analysis: Theory, Methodology and Applications*, forthcoming.

Conte, S.D., H.E. Dunsmore and V.Y. Shen (1986), *Software Engineering Metrics and Models*, Menlo Park, CA: Benjamin/Cummings Pub. Co.

Elam, J. (1991), "Evaluating the Efficiency of IS Organizations using Data Envelopment Analysis," *Proc. of International Function Point Users Group 1990 Fall Conference*, October 1991.

Fenton, N.E. (1991), *Software Metrics: A Rigorous Approach*, London: Chapman and Hall.

Fenton, N.E. (1994), "Software Measurement: A Necessary Scientific Basis," *IEEE Transactions on Software Engineering*, 20, 3.

Ferdinand, A.E. (1993), *Systems, Software and Quality Engineering*, NY: Van Nostrand Reinhold.

Fried, H.O., K.C.A. Lovell and S.S. Schmidt (editors) (1993), *The Measurement of Productive Efficiency*, NY: Oxford University Press.

Gibbs, W.W. (1994), "Software's Chronic Crisis", *Scientific American*, September 1994.

Jeffery, D.R. and G. Low (1990), "Generic estimation tools in management of software development, *Software Engineering Journal*, 5, 4.

Jeffery, D.R. (1987), "Time-sensitive cost models in commercial MIS environment," *IEEE Transactions on Software Engineering*, 13, 7.

Jones, C. (1986), *Programming Productivity*, New York, NY: McGraw-Hill.

Jones, C. (1991), *Applied Software Measurement*, New York, NY: McGraw-Hill, Inc.

Kemayel, L., A Mili and I. Ouederni (1991), "Controllable Factors for Programmer Productivity: A Statistical Study", *Journal of Systems Software*, 16.

Kemerer, C.F. (1987), "An Empirical Validation of Software Cost Estimation Models", *Communications of the ACM*, 30, 5.

Kitchenham, B. and N.R. Taylor (1984), "Software Cost Models," *ICL Technical Journal*, 4, 1.

McCall, J.A., P.K. Richards and G.F. Walters (1977), "Factors in Software Quality, Vols I, II, III," *US Rome Air Development Center Reports* NTIS AD/A-049 014, 015, 055.

Norman, M. and B. Stoker (1991), *Data Envelopment Analysis: The Assessment of Performance*, Chichester, England: John Wiley and Sons, Ltd.

Putnam, L.H. and W. Meyers (1992), *Measures for Excellence: Reliable Software on Time, Within Budget*, NJ: Yourdon Press.

Seiford, L.M. and R.M. Thrall (1990), "Recent Developments In Dea The Mathematical Programming Approach To Frontier Analysis," *Journal Of Econometrics*, 46.

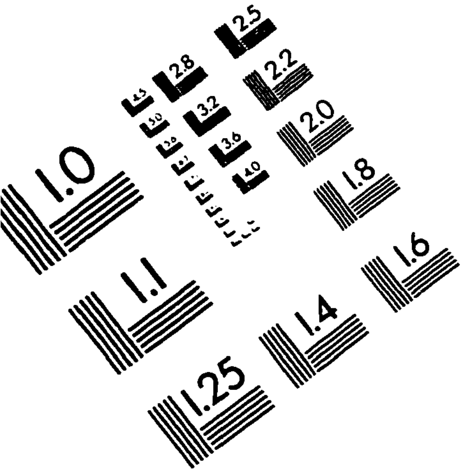
Thompson, R.G., L.N. Langemeier, L. Chih-Tah, L. Euntaik and R.M. Thrall (1990), "The Role of Multiplier Bounds in Efficiency Analysis with Application to Kansas Farming," *Journal of Econometrics*, 46.

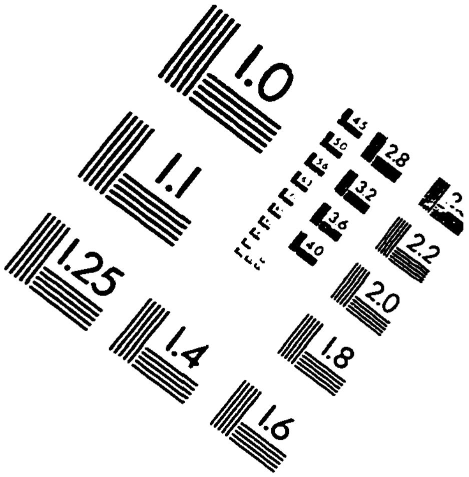
Weber, M. and K. Borcherding (1993), "Behavioral Influences on Weight Judgements in Multiattribute Decision Making," *European Journal of Operational Research*, 67.

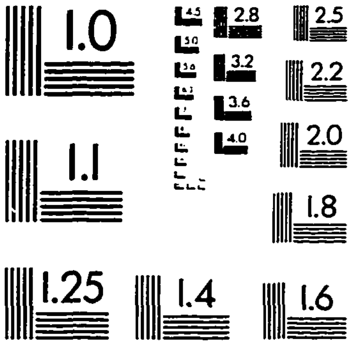
Wong, Y.-H.B., and J.E. Beasley (1990), "Restricting Weight Flexibility In DEA," *Journal of the Operational Research Society*, 41, 9.

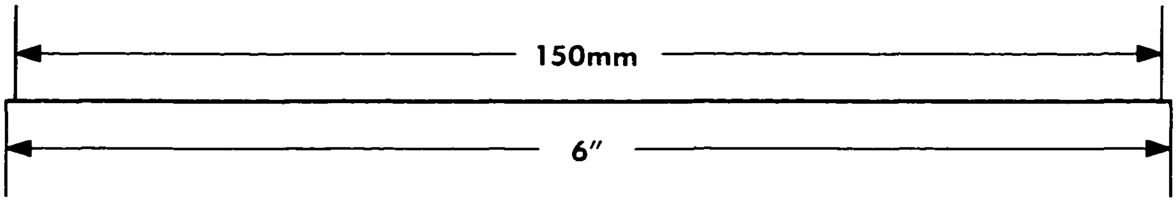
Wortman, D.B. (1994), "Software Engineering", CSC 2105 Lecture Notes, Department of Computer Science, University of Toronto.

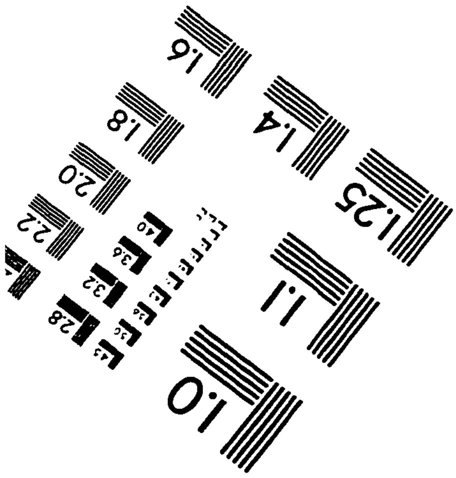
IMAGE EVALUATION TEST TARGET (QA-3)

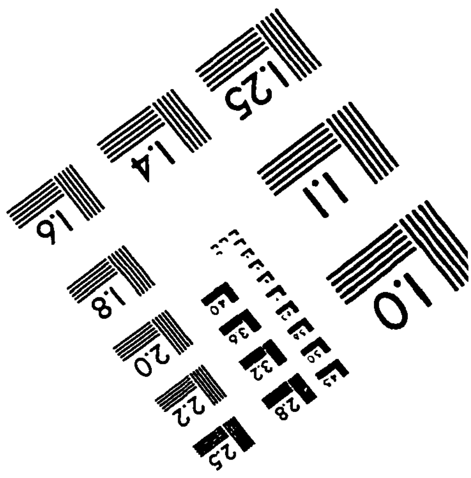












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/286-5989