

2004

Technology Forecasting Using Data Envelopment Analysis

Oliver Lane Inman
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Operations Research, Systems Engineering and Industrial Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Inman, Oliver Lane, "Technology Forecasting Using Data Envelopment Analysis" (2004). *Dissertations and Theses*. Paper 2682.

<https://doi.org/10.15760/etd.2678>

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

TECHNOLOGY FORECASTING
USING DATA ENVELOPMENT ANALYSIS

by
OLIVER LANE INMAN

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY
in
SYSTEMS SCIENCE: ENGINEERING AND TECHNOLOGY MANAGEMENT

Portland State University
2004

Dedication

I would like to dedicate this thesis to my grandfather Jack H. Inman who was not able to see me complete this seemingly never ending adventure, and will sadly be missed.

Acknowledgments

In every great journey, everyone depends on someone to get them through their trials and tribulations. I have been blessed with an array of friends and relatives who have endured my fits and triumphs over the last seven years. I would like to specifically acknowledge Kristi Cavanaugh, who saw me through the beginning, Claire Preaud who dealt with me through the middle, and my parents Jack and Aggie who have dealt with me from the very beginning. I would also like to acknowledge Mr. Porter for providing me with the data required for my third case study and specifically thank my committee members without which I would have never finished. Particularly Dr. Anderson, Dr. Wakeland, and Dr. Milosevic who through their patience and guidance helped me accomplish this work.

Table of Contents

Acknowledgments.....	ii
Table of Contents.....	iii
List of Tables.....	ix
List of Figures.....	x
Glossary.....	xi
I. Introduction.....	1
A. Introduction to the Study.....	1
B. Problem Statement.....	2
C. Research Objective.....	5
D. Research Process.....	6
E. Major Contributions of the Research.....	7
F. Overview of Dissertation.....	8
II. Literature Review.....	10
A. General Technology Forecasting Literature.....	11
1. Monitoring.....	13
a. Focus and Objective Determination.....	14
b. Technology Description and Pertinent Context Mapping.....	14
c. Adaptation of Proper Monitoring Strategy.....	14
d. Communication and Interpretation of the Results.....	15
2. Expert Opinion.....	15
a. Selecting Experts.....	15
b. Methods of Collecting Expert Opinion.....	16
3. Trend Extrapolation.....	18
a. Naïve Models.....	20
b. Adaptive Weighting.....	22
c. Trend Analysis.....	23
d. Substitution Curves.....	25
e. Linear and Multiple Linear Regression.....	25
4. Modeling and Simulation.....	26
5. Scenarios.....	28
6. Critical Review of General Technological Forecasting Literature.....	29
a. Accounting for Characteristic Interrelationships.....	30
b. Accounting for External Influences.....	30
c. Accounting for Changing Preference	31
d. Requiring Historical Data.....	31

e. Prone to Expert Error.....	32
f. Adapting to Disruptive Technologies.....	32
g. Predicting Future Characteristics.....	33
h. Summary.....	33
B. State-of-the-Art and Trade-off Surfaces Literature.....	34
1. General Procedure.....	35
2. Scoring and Regression.....	37
3. Dodson’s Approach.....	40
4. Technometrics.....	41
5. Critical Review of SOA and Trade-off Surface Literature.....	42
a. Emphasizing the Best Available Technologies.....	42
b. Allowing Trade-off Change.....	44
c. Allowing New/Old Characteristics.....	45
d. Predicting Future Characteristics.....	45
e. Handling Correlated Attributes.....	46
f. Handling Multiple Outputs.....	46
g. Handling Variable Returns.....	47
h. Summary.....	48
C. Data Envelopment Analysis Literature.....	49
1. Introduction.....	50
2. Weight Restrictions.....	53
3. Malmquist Productivity Indices.....	54
4. Strengths and Limitations of DEA.....	56
5. Procedure for Implementing DEA.....	56
a. Decide Purpose.....	57
b. Define DMU.....	57
c. Determine Inputs/Outputs.....	57
d. Select DEA Model.....	58
e. Collect Data.....	61
f. Perform Analysis.....	61
g. Examine Results.....	61
h. Improve Model.....	62
6. Example of Using DEA to Evaluate Technology.....	62
a. Decide Purpose.....	63
b. Define DMU.....	63
c. Determine Inputs/Outputs.....	63
d. Select DEA Model	64

e. Collect Data	64
f. Analyze Technological Progress.....	65
g. Examine Results.....	66
h. Improve Model.....	67
7. Critical Review of DEA Literature.....	68
a. Allowing for a Single DMU with Only One Observation	69
b. Allowing Irregular Time Periods.....	70
c. Summary.....	70
D. Critical Review of Literature Gaps.....	70
1. Extrapolative Technological Forecasting with Trade-Off Surfaces.....	71
a. Emphasizing the Best Technologies.....	72
b. Dynamic Nature of Trade-off Surfaces.....	73
c. Handling Correlated Characteristics.....	74
d. Predicting Future Technologies.....	75
2. Data Envelopment Analysis.....	75
3. Literature Gap Summary.....	77
III. Research Methodology and Model Development.....	78
A. Purpose.....	78
B. Methodology Overview.....	78
C. Research Objective	79
D. Research Approach.....	79
E. Using DEA to Represent the SOA.....	81
1. Determine Scope of Forecast.....	82
2. Define a Product	83
3. Define SOA Characteristics.....	83
4. Determine DEA Model.....	84
a. Orientation.....	85
b. Returns to Scale.....	85
c. Virtual Product Projection.....	86
5. Collect Data.....	86
6. Analyze Technological Progress.....	87
a. Mapping Technological Progress.....	87
b. Time Considerations.....	89
c. Forecasting Future Technologies.....	90
7. Examine Results.....	91
F. Stepping Through the Process	92
1. Determine the Scope of the Forecast.....	92

2. Define a Product.....	93
3. Define SOA Characteristics.....	93
4. Determine DEA Model.....	93
5. Collect Data	93
6. Analyzing Technological Progress.....	94
a. Mapping Technological Progress.....	94
b. Forecasting Future Technologies.....	97
7. Examining Results.....	98
8. Improving the Model	100
G. Model Construction.....	103
H. Model Verification.....	104
1. Fixed Rate of Change.....	104
2. Normally Distributed Rate of Change.....	105
IV. TFDEA Validation Case Studies.....	107
A. Database System Performance: A Two Dimensional Case.....	107
1. Determine the Scope of Forecast.....	108
2. Define a Product	108
3. Define SOA Characteristics.....	109
4. Determining the Specific Model	109
a. Orientation.....	110
b. Returns to Scale.....	110
5. Collect Data.....	110
6. Analyze Technological Progress.....	111
a. General Results.....	111
b. Comparison Regression Results.....	118
7. Analysis of Results.....	120
a. Double Orientation.....	120
b. Identification of Disruptive Technologies.....	120
c. Effectiveness of the Forecast.....	121
d. Comparison to Other Methods.....	121
B. Microprocessors: A Multiple Dimensional Example.....	121
1. Determine Scope of Forecast	122
2. Define a Product	122
3. Define SOA Characteristics.....	123
4. Determine DEA Model	126
a. Orientation.....	126
b. Returns to Scale.....	126

5. Collect Data	126
6. Analyze Technological Progress.....	127
a. General Results.....	127
b. Comparison to Moore's Law.....	128
7. Analysis of Results.....	130
a. Dataset Size	130
b. Effectiveness of the Forecast.....	130
c. Comparison with Other Methods.....	130
C. Persistent Storage: Application to Multiple Technological Approaches.....	131
1. Determine Scope of Forecast.....	131
2. Define a Product.....	131
3. Define SOA Characteristics	132
4. Determine DEA Model.....	135
a. Orientation.....	135
b. Returns To Scale.....	135
5. Collect Data.....	135
6. Analyze Technological Progress.....	136
7. Analysis.....	139
a. Identification of Unrealistic Expectations.....	139
b. Expansion of Scope.....	140
c. Identification of Disruptive Technologies.....	140
d. Effectiveness of the Model.....	140
e. Comparison to Other Techniques.....	141
D. Analysis of Results.....	141
1. Effectiveness of the Model.....	141
a. Using DEA to Represent SOA and Technology Trade-off Surfaces.....	142
b. Extension of Temporal DEA to Monitor Technology.....	142
c. Extension of Temporal DEA to predict future technologies.....	143
2. Comparison to Other Techniques.....	144
V. Conclusions.....	146
A. Research Overview.....	146
B. Results of Case Studies.....	151
1. OLTP Databases.....	151
2. Moore's Law Reexamined.....	152
3. Persistent Storage Reexamined.....	153
4. Discussion of Results.....	153
C. Research Contributions.....	155

1. Method Using DEA to Represent the SOA.....	156
2. Method for Monitoring SOA Advancement.....	157
3. Method to Forecast Future SOA.....	157
4. Linking DEA to Technology Forecasting.....	158
D. Management Implications.....	159
1. Requirements.....	159
a. Data.....	159
b. Understanding of Technology Structure.....	160
2. Strengths.....	160
a. Focusing on Best Technology.....	160
b. Ability to Forecast Future Characteristics.....	161
c. Ability to Estimate the Availability of Future Characteristics.....	161
d. Ability to Evaluate Multiple Technical Approaches Simultaneously.....	161
3. Limitations.....	161
a. Sensitivity to Disruptive Technologies.....	161
b. Assumption of Constant Rates of Change	162
4. Summary.....	162
E. Future Research.....	163
1. Related to Data Envelopment Analysis.....	164
a. Identification of Sample Size Requirements.....	164
b. Scale Efficiency.....	164
c. Variable Rates of Change.....	165
d. Analysis of the Rate of Change.....	165
2. Related to Technology.....	165
a. Examination and Comparison of Orientation.....	165
b. Further Studies with Additional Technologies.....	166
c. Integration with Technological Approach Forecasting.....	166
d. Integration with Expert Opinion.....	166
e. Integration with Preference Structure.....	167
f. Usage of Multiplier Model to Identify Primary Technology Drivers.....	167
3. Identifying Future Markets or New Targets.....	167
Appendix A - TFDEA Crib Sheet.....	179
Appendix B - TPC Dataset.....	181
Appendix C - Moore's Law Dataset.....	186
Appendix D - Software Components.....	187

List of Tables

Table 1 - Most Common Reasons for Technological Forecasting [80].....	1
Table 2 - Major Contributions of Research.....	8
Table 3 - Attributes Most Frequently Addressed by Technological Forecasts [81].....	11
Table 4 - Technological Forecasting Methodologies [78].....	12
Table 5 - Objectives for Monitoring Technology [27].....	13
Table 6 - Ideal Characteristics of Experts [64].....	16
Table 7 - Disadvantages of Committees [70].....	17
Table 8 - Research Issues in Technology Forecasting.....	30
Table 9 - Current Issues with SOA Literature.....	43
Table 10 - Strengths and Limitations of DEA.....	56
Table 11 - Common DEA Models Available.....	61
Table 12 - Sample Dataset of USB Pen Drives.....	65
Table 13 - USB Pen Drive Output-Oriented CRS DEA Efficiencies.....	66
Table 14 - Current Issues with Temporal DEA.....	69
Table 15 - Sample USB Pen Drive Data.....	94
Table 16 - 2003 USB Pen Drive VRS SOA Scores.....	100
Table 17 - 2003 Output Oriented Rate of USB Pen Drive Technological Progress.....	102
Table 18 - TPC Results of ROC Validation with Output-Oriented Model.....	112
Table 19 - TPC Results of ROC Validation with IO-OO Output-Oriented Model.....	113
Table 20 - TPC Annualized VRS SOA ROC for 03/17/2004.....	114
Table 21 - Summary of Upcoming Database Releases Based on 2004-03-17 Results....	115
Table 22 - 2005-03-17 TPC VRS SOA Predictions Based on 2004-03-17 Data.....	117
Table 23 - TPC Regression Results 09/15/2000 – 02/12/2003.....	119
Table 24 - Microprocessor Results of Validation with Output-Oriented Model.....	127
Table 25 - Results of Null Hypothesis Test for DEA vs. Moore's Law.....	129
Table 26 - Persistent Storage Validation Results with Output-Oriented Model.....	137
Table 27 - Persistent Storage Validation Results with IO-OO Output-Oriented Model .	138
Table 28 - Research Contributions.....	155

List of Figures

Figure 1 - Flow of Dissertation.....	7
Figure 2 - Literature Review Overview.....	10
Figure 3 - Technology Envelope Built by Technological Approaches [26].....	20
Figure 4 - Number of Transistors on Intel Microprocessors Versus Moore's Law [55]....	22
Figure 5 - Advancing Technology Frontier.....	37
Figure 6 - Techniques to Model the SOA.....	49
Figure 7 - Procedure for Implementing DEA [6].....	56
Figure 8 - Graphic Representation of Input vs. Output-Oriented.....	58
Figure 9 - Graphical Representations of Returns to Scale.....	59
Figure 10 - USB Pen Drive Technology Model.....	64
Figure 11 - USB Pen Drive Efficiency Frontier.....	65
Figure 12 - USB Pen Drive Output-Oriented VRS Frontier.....	67
Figure 13 - Applications in Technology Evaluation with DEA.....	69
Figure 14 - Technology Forecasting Research Issues Mapped to Gaps.....	72
Figure 15 - DEA Research Issues to DEA Gap.....	77
Figure 16 - Research Methodology.....	78
Figure 17 - Research Design.....	80
Figure 18 - Map of Combined DEA and SOA Approach.....	82
Figure 19 - Procedure for Mapping Technological Change.....	89
Figure 20 - 2001 USB Pen Drive Output Oriented VRS Frontier.....	95
Figure 21 - 2001 and 2002 USB Pen Drive Output-Oriented SOA Frontiers.....	95
Figure 22 - 2003 Predicted USB Pen Drive SOA Using 2002 Data.....	98
Figure 23 - 2003 USB Pen Drive Prediction and Products Using 2002 Data.....	99
Figure 24 - 2003 USB Pen Drive Frontier Projected with IO and OO.....	100
Figure 25 - 2003 USB Pen Drive SOA Frontier with All Products.....	101
Figure 26 - Predicted 2004 VRS SOA from Previous Data with both Orientations.....	102
Figure 27 - Basic OLTP Technology Model.....	109
Figure 28 - Annualized Rates of Change for TPC-C.....	114
Figure 29 - 2005-03-17 TPC SOA Forecast Using 2004-03-17 Data.....	117
Figure 30 - Enhanced Microprocessor Technology Model.....	123
Figure 31 - Microprocessor Rates Of Change for Output-Oriented Model.....	128
Figure 32 - Persistent Storage Functional and Structural Model.....	132
Figure 33 - DISK/TREND Rates of Change for Output-Oriented Model.....	137
Figure 34 - Research Design.....	151

Glossary

Absolute Efficiency:	Corresponds directly to the overall efficiency of a machine or observation in converting inputs to outputs.
Clock Speed:	The number of cycles per second at which a processor may execute a minimal instruction set.
DEA:	Data Envelopment Analysis.
Die Size:	Amount of surface area required by an integrated circuit or microprocessor [5].
DMU:	Decision Making Unit. “Generically a DMU is regarded as the entity responsible for converting inputs into outputs and whose performances are to be evaluated [18].” In this work, DMUs refer to individual products of a technology which are being evaluated.
DRS:	Diminishing Returns to Scale.
Efficiency:	The effectiveness of a DMU in converting given inputs into outputs. This is primarily based on Farrell’s concept of production efficiency [43].
Functional Characteristics:	Those characteristics which measure performance to the end user [61].

Input Orientation:	Input Oriented DEA attempts to minimize input required to produce a given set of output. This is akin to traditional hedonics.
IRS:	Increasing Returns to Scale.
Minimum Feature Size:	The smallest sized feature producible for integrated circuits.
MIPS:	Millions of instructions per second. Early method used to measure computer performance. Thought by many to be an extremely limited measure of computing performance [99].
Observation:	An observation in DEA is a single observation of a DMU. In the event that a single DMU may change in relative efficiency and structure over time, multiple observations may be made. In the case of technology forecasting, performance of a product does not change over time for a release, thus an observation is effectively defined as a DMU or product.
Output Orientation:	Provided a given set of inputs, DEA attempts to maximize the possible outputs.
Product:	A product is a device or machine using technology to

convert structural inputs to functional outputs. It is from products that the technology used is to be predicted.

- Proxy DMU: A virtual DMU constructed to represent a non-efficient DMU on the efficiency frontier. This is similar to weighting regression results.
- RDBMS: Relational Database Management System.
- Relative Efficiency: Efficiency of observations relative to all other observations; this is not to be confused with absolute efficiency.
- SOA: State of the Art is the “state of best implemented technology as reflected by the physical and performance characteristics actually achieved by the physical and performance characteristics actually achieved during the time period in question [31].”
- SPEC: Standard Performance Evaluation Corporation.
Organization whom’s mission is “To establish, maintain, and endorse a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems [99]”
- SPECInt: Standard Performance Evaluation Corporation Integer

	Benchmarks. A suite of benchmarks designed to evaluate the integer performance of processors [99].
SPECFP:	Standard Performance Evaluation Corporation Floating Point Benchmarks. A suite of benchmarks designed to evaluate the floating point performance of processors [99].
Structural Characteristics:	Those characteristics of a product required to provide the function of the product. Introduced as “technical parameters” by Alexander and Nelson [1].
Technical Approach:	Specific means of solving a problem or performing a particular function [70].
Technological Change:	Change in the effectiveness of technology over time. In events where decreasing structural requirements provides more functional performance, it is called technological growth. In events when more structure provides less function technology is said to decrease.
Technology forecasting:	“A prediction of the future characteristics of useful machines, procedures, or techniques [70].”
Technological Index:	An index composed of numerous factors indicating the level of technology implemented within a product. Used

by Knight [60], Alexander [1], Dodson [31], Martino [68], and others it aims to address the limitations of single attributes.

- TPC: Transaction Processing Performance Council.
Organization established to measure the effectiveness of relational database management systems. “The TPC defines transaction processing and database benchmarks and delivers trusted results to the industry” [103]
- tpmC: Transactions per minute C. The number of transactions per minute using the TPC – C benchmark suite.
- VRS: Variable Returns to Scale

I. Introduction

A. Introduction to the Study

The ability to anticipate future capabilities of technology products has broad implications for organizations. Betz asserts, “The design of products to beat future products of competitors is the fundamental reason why technology forecasting is vital to new product development [14].” Other reasons for technological forecasting are summarized by Porter as listed in Table 1.

Table 1 - Most Common Reasons for Technological Forecasting [80]

● Guidance of resource allocation
● Identification of market opportunities or threats
● Guidance of staff, facilities, or capital planning
● Development of strategies or policies
● Assistance with R&D management
● Evaluation of new products
● Maximize gain or minimize loss due to internal or external elements of the organization

Technological forecasting permits management to allocate resources better based on anticipated technological trends. If an organization can estimate the future capabilities of technology, then it can better allocate resources, guide staff, or perform facilities and capital planning. They may also identify potential new markets and opportunities to exploit a given technology beyond its originally intended purposes. New products can be more readily assessed as to their impact by comparing them with previously introduced products. This permits organizations to understand the state of new technology.

Modern technology forecasters use an array of methods to predict the future performance of a technology, including methods based on complex mathematics such as time-series analysis, stochastic methods, and simulation. These methods often rely on the assumption that past behavior will continue. These forecasts compliment techniques based on expert opinion and panels by providing extrapolative results that are quantified and reproducible. Although forecasters attempt to make accurate forecasts, insights gained from the technological forecasting process can provide value whether or not the predictions are accurate [8][70][81]. In summary, modern forecasters have an array of flexible tools that may be used for a number of business purposes. Although the forecasts may not always be accurate, the insight they help to generate can be valuable and have significant impact on their organizations. This particular area of research is significant in its ability to help organizations avoid costly mistakes. For the reasons presented this is a significant area of research.

B. Problem Statement

For quantifiable forecasts, time series analysis and related tools are frequently used to forecast technology. However, these tools are often insufficient because, although many technology characteristics are quantifiable, it is difficult to determine which ones accurately represent the “level” of technology being forecast. One example explored further in this research is Moore's law, which attempts to represent the technology of

integrated circuits (ICs) with the number of IC transistors produced at minimal cost [73].

Although transistor count is an important characteristic, a more comprehensive measure would take into account additional features [5][7]. Single variable trend extrapolation fails to account for multiple technological characteristics and their functional trade-offs.

To deal with multiple characteristics and their trade-offs, forecasters use multidimensional trade-off surfaces to represent the state of technology available at a point in time. One common method to create these surfaces is that of multiple linear regression. Since this approach leverages central-point tendencies, the resulting formulation does not represent the best that is achieved. Scoring models may offer the most useful technology, but do not tend to measure the “best.” This gap is summarized below.

Gap #1:	<i>Current extrapolative forecasting methodologies do not address the “best” available technology but an aggregate of all technologies available.</i>
----------------	---

In addition, regression does not assume any variation in the objective function, and requires attribute independence reducing the number of eligible attributes. Scoring models allow for more focused applications, however they fail to take into account the potential for changing preferences over time. Such approaches do not account for the dynamic nature of trade-offs faced by designers, product managers, or engineers when designing products. This gap is summarized below.

Gap #2:	<i>Current extrapolative forecasting methodologies do not take into account the dynamic nature of trade-off surfaces.</i>
----------------	--

Furthermore, fixed constraints limit the forecaster's ability to anticipate or recognize early a potentially disruptive technology [26]. Another shortcoming of the present methodologies is that they require independence of important attributes. For less understood technologies, it may be difficult to completely isolate attributes from each other. This gap is summarized below.

Gap #3:	<i>Current extrapolative forecasting methodologies require non-correlated attributes.</i>
----------------	--

Finally, current methodologies only allow for one output at a time to be analyzed regardless of method. Popular scoring and regression models aggregate all attributes to a single technology score which is absolute and unwavering. The problem with this is that there may be a number of output and input interrelationships that may not be represented by such a simplistic model. This gap is summarized below.

Gap #4:	<i>Current extrapolative methods are limited to a single output that can not be disaggregated.</i>
----------------	---

One methodology that can address these gaps is data envelopment analysis (DEA). DEA provides a flexible means to represent technology, through a well-established productivity measuring methodology that has been cited in over 1500 references [94]. By using DEA to measure technology, it is possible to identify the rates of technological

change without the assumption of fixed trade-offs or complete attribute independence.

Using DEA, it is also possible to measure multiple inputs and outputs simultaneously to directly determine the most efficient technologies. However current DEA methods do not allow for single individual decision making units (DMUs) to be present only once over time.

Gap #5:	<i>Current temporal models of DEA do not allow for DMUs which are observed at only one time period and introduced at irregular intervals.</i>
----------------	--

C. Research Objective

Based on the gaps discussed there is a need for better extrapolative techniques to forecast technologies.

Research Objective:	<i>Develop a methodology for technology forecasting which provides for a readily calculable method to measure the SOA and its advance by extending current temporal DEA to allow for DMUs which are introduced once at irregular intervals.</i>
----------------------------	--

The strengths of DEA provide a potential method to address many shortcomings in the current extrapolative technology forecasting techniques. In order to assess technological change, it is necessary to address gaps #1-#4 as indicated in research question #1 below.

Research Question #1:	<i>How can DEA be used to measure the SOA and trade-off surfaces?</i>
------------------------------	--

This provides forecasting practitioners a method to represent the SOA trade-off surface for a technology. Once the surface is understood there is a need to track the change over time. This can be done by addressing gap #5 and is represented in research question #2.

<i>Research Question #2:</i>	<i>How can temporal DEA be extended to monitor technology of products that are only observed only once at irregular intervals?</i>
-------------------------------------	---

With the rate of change ascertained, forecasters can use the insights provided by research question #2 to determine the characteristics of future products as described by research question #3 below.

<i>Research Question #3:</i>	<i>How can temporal DEA be used to forecast future attributes of technological products?</i>
-------------------------------------	---

D. Research Process

This research creates and validates a new methodology technology forecasting with DEA (TFDEA) and is conducted in the first three stages listed in Figure 1. The first stage, is a literature review that examines current practices for technology forecasting. Additional research reviews the current DEA literature to provide a background for the second stage. The second stage uses DEA to address the discussed gaps in current extrapolative technology forecasting methodologies. This stage is completed with sample data for model verification. The third stage involves application to three subjects.

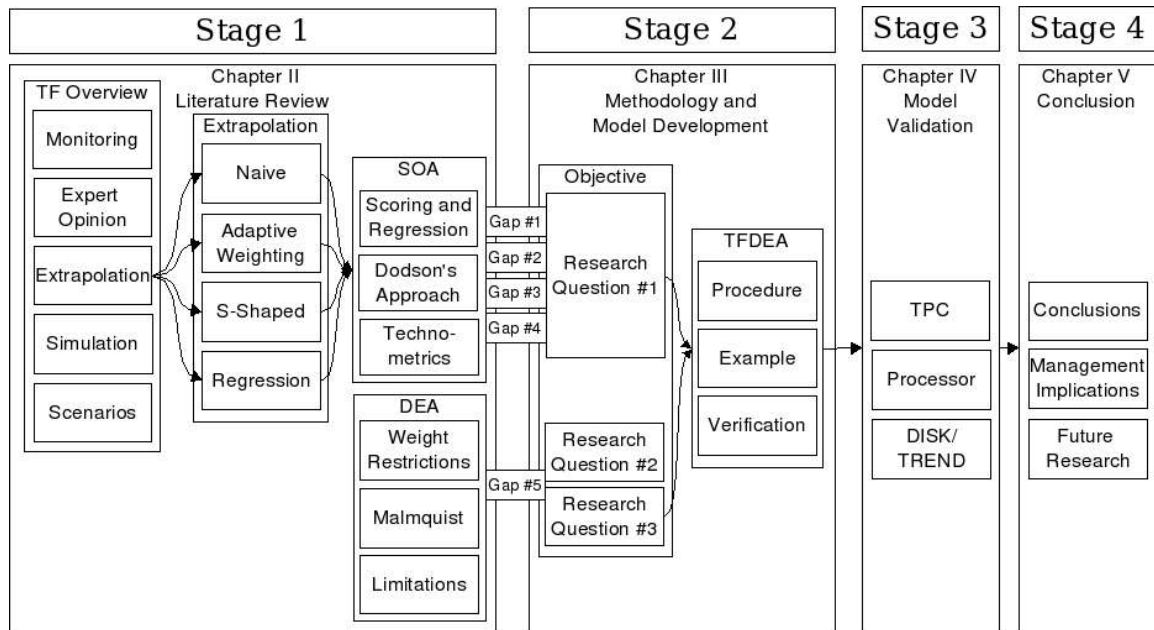


Figure 1 - Flow of Dissertation

Each application addresses both research questions by examining the validity of the proposed methodology. The first application studies online transaction processing systems (OLTP) to provide a straightforward easily visualized two dimensional model. The second application is a reexamination of, and a six dimensional expansion of, one of the best known forecasts of the twentieth century: Moore's Law. The final application is applied to a subset of the DISK/TREND data discussed in Clayton Christensen's *Innovator's Dilemma*, thus providing a link to popular management forecasting literature and expanding the methodology to multiple technical approaches.

E. Major Contributions of the Research

The overall objective of this research is to provide practitioners with a practical and

robust new methodology to monitor technological progress over time, and use the insights gained to forecast future product capabilities. This will be done by addressing the five gaps presented, and answering the two research questions. By providing a more robust means of forecasting to practitioners, it will provide a new approach to setting future benchmarks and realistic expectations for future products. In the event that a product is truly ground breaking, it should also provide organizations with this insight. In addition, technology forecasters may use the already rich arena of DEA research to resolve additional forecasting issues. The major contributions which this research provides are listed in Table 2.

Table 2 - Major Contributions of Research

Contribution 1: A new temporal DEA model, extended to monitor technology of DMUs that are only observed once at irregular intervals.

Contribution 2: A new methodology for forecasting multiple product capabilities.

Contribution 3: Linking DEA and technology forecasting

F. Overview of Dissertation

Chapter 2 provides an overview of quantitative and non-quantitative methods used for technological forecasting. In addition, it discusses the origins of the concept of SOA and a critical review. Following this review is an overview of DEA and its time based methods as well as the relevant literature on the subject. Chapter 3 discusses the research methodology and develops a new forecasting methodology known as technology

forecasting with DEA (TFDEA). Chapter 4 includes a set of applications used to demonstrate and validate the proposed model. The final chapter presents conclusions, future research, and management implications that may be drawn from this research as presented in Figure 1.

II. Literature Review

This chapter consists of four sections illustrated in Figure 2. The first section presents a general overview of technology forecasting as classified by Porter *et al* [81]. This is followed by a discussion of measuring technology with trade-off and state-of-the-art (SOA) surfaces. The third section describes data envelopment analysis (DEA) and how it may be used to evaluate technology. The final section includes a gap analysis of the current literature for extrapolative technology forecasting for the state of the art and the

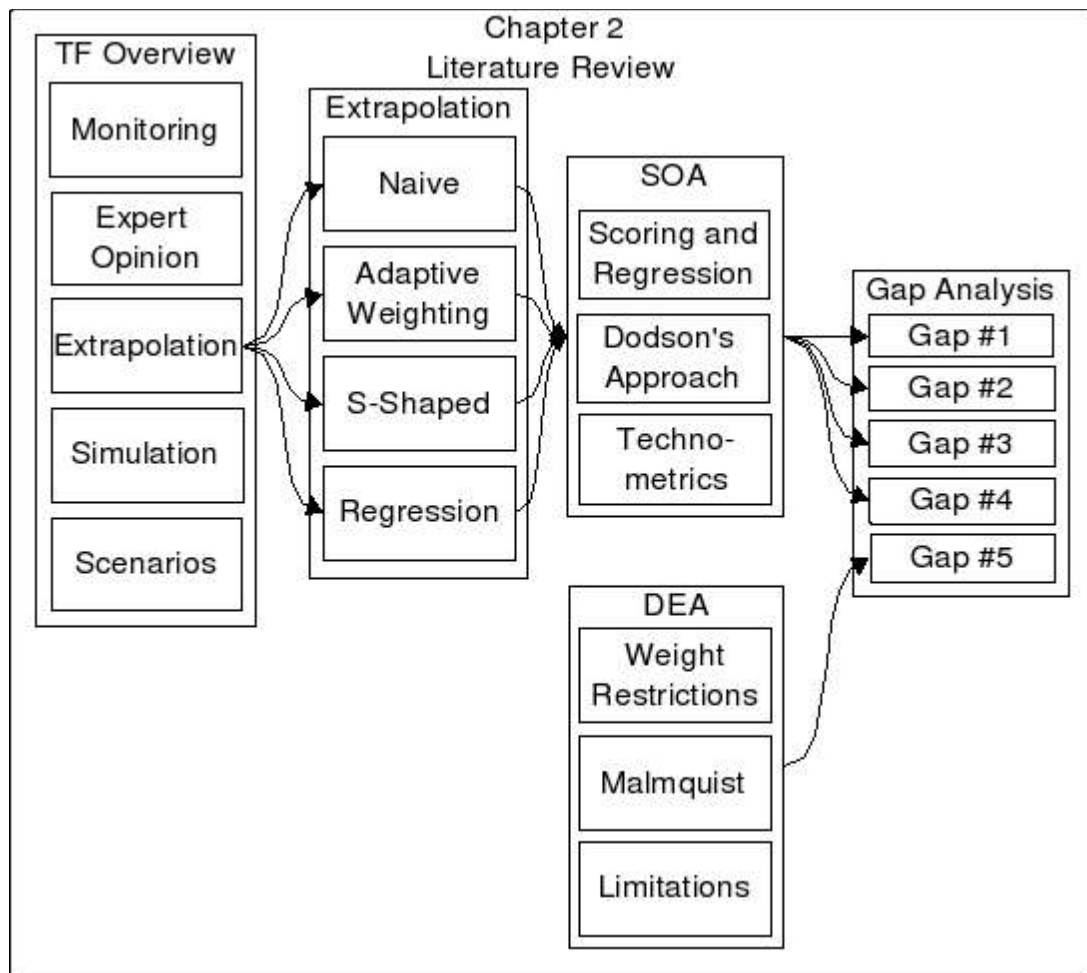


Figure 2 - Literature Review Overview

gap in current temporal DEA methodologies.

A. General Technology Forecasting Literature

In 1971, Martino defined technology forecasting as a means “to predict the future characteristics of a useful machine [69].” In this context, the term machine refers not only to physical devices, but tools, techniques and procedures that provide some function to an end-user. Technology may be defined as a means of combining structural components to deliver a set of functions to a user [1][60]. According to Porter *et al.* those attributes of technology most frequently forecast are summarized in Table 3.

Table 3 - Attributes Most Frequently Addressed by Technological Forecasts [81]

● Functional capability
● Market penetration
● Diffusion of technology to other sectors
● Likelihood and timing of technological breakthroughs

The forecasting of technology involves many types of causal elements including but not limited to social, economic, or technological, but it has not attained a set of given laws and rules. This is due to the inherent complexity of the systems that are forecast and the difficulty of removing them from their technical, social, political, environmental, economic and ethical contexts [62]. These factors greatly affect the growth, adoption and diffusion of technology. In the jargon of some scientific disciplines, these are dirty problems for which it is virtually impossible to design series of verifying experiments [81]. There is also the possibility that forecasting biases may influence the methodology and data type used to match *a priori* assumptions (bias) [81].

Typically, the focus of forecasts is on a single technology or family of technologies.

The predominant methods such as trend extrapolation, expert opinion and scenarios often identify outcomes without accounting for the characteristic interrelationships [81].

Primarily, forecasts are categorized as extrapolative or normative [70][81][108].

Extrapolative forecasts predict the future of a technology from historical data under the assumption that the future trends will mimic the past. This data is then used in conjunction with extrapolative techniques such as time series or regression to determine the future of the functional and structural characteristics of the technology being forecast.

Normative forecasting assumes a future in a given state and looks backward to determine the necessary developments to reach it. Often extrapolative forecasts are used to generate normative forecasts that examine external issues that must be considered. Table 4 lists classifications of forecasting by Rossini and Porter [78], which are explained in the following sections. With the exception of scenarios, these methods are predominantly considered extrapolative.

Table 4 - Technological Forecasting Methodologies [78]

● Monitoring
● Expert Opinion
● Trend Extrapolation
● Modeling and Simulation
● Scenarios

1. Monitoring

Coates proposes that monitoring is “to watch, observe, check, and keep up with developments, usually in a well-defined area of interest for a very specific purpose [27].”

Monitoring is not forecasting *per se*; however, it is often necessary to perform valid monitoring prior to future forecasting activities. It may be used as a one-time event or a series of ongoing exercises. Point-in-time monitoring offers insight into the environment for discovery missions. Examples of monitoring include bibliometrics and patent searches [31][57][70][74][75][79][80].

As its name implies, monitoring deals primarily with identifying key technological attributes and gauging their progress. Once a proper understanding of the technology is achieved, it is possible to move forward with other methodologies. Possible objectives for monitoring of technologies are listed in Table 5.

Table 5 - Objectives for Monitoring Technology [27]

- | |
|--|
| ● Detecting important scientific, technical, or socioeconomic events |
| ● Defining potential threats for the organization |
| ● Seeking opportunities for the organization |
| ● Alerting management to trends that are converging, diverging, speeding up, slowing down or interacting |

There are a number of means to perform monitoring including bibliometric studies that may permit preemptive discovery of breakthroughs. These methods are hard to quantify but often require expert analysis to determine those technologies or elements

that have future effects on the current objectives [70][81]. The overall process for monitoring summarized by Porter is presented below [81]. It may be used as a foundation for all technology forecasting methodologies.

a. Focus and Objective Determination

Specific objectives must be explicitly specified to best target monitoring resources. Agreement should be reached amongst those involved concerning the scope, extent, personnel, and whether the process will be ongoing. Focus is of utmost importance as its lack tends to generate overly large amounts of indiscernible data [70].

b. Technology Description and Pertinent Context Mapping

A description of the technology to be monitored and the mapping of related technologies should be performed prior to any forecast. In addition, the pertinent technological system, critical milestones, and vital socio-economic influences should be identified. With ongoing monitoring efforts, these elements may be reexamined using the data acquired through the monitoring process.

c. Adaptation of Proper Monitoring Strategy

Monitoring approaches are typically differentiated by familiarity with the technologies being monitored. Porter suggests three levels of familiarity: hot, warm, and cold [81]. When little is known about a technology, a better strategy is to map out the technology and identify its key elements and impacts. As familiarity increases, this role changes to

monitoring understood interrelationships.

d. Communication and Interpretation of the Results

There is little use for data if it is not presented to those who have a vested interest in that data. Establishing communication channels to interested parties is important to relay the discoveries to the interested parties. Typical tools to assist in this communication include technology maps, milestones, and communications infrastructure.

2. Expert Opinion

Expert opinion is of particular interest for the early stages of technologies, when little empirical data is available to extrapolate trends or build complete models. Other situations that merit expert opinion include occasions where external factors such as public opinion, political motivations, ethical, or moral considerations dominate economic and technical considerations. For this reason, it is important that the forecaster limit the scope of the forecast for the experts. There have been criticisms of using “experts” for forecasting including the difficulty of finding good experts [10][70]. Much of this can be overcome with proper guidance.

a. Selecting Experts

Lipinski and Loveridge [64] suggest that when selecting experts, one should target the characteristics presented in Table 6. Selection of experts should be based on peer review and subject matter understanding with respect to the above three parameters. In the event

that organizational resources are not available, it may be necessary to go to outside sources of expertise determined by *Who's Who* listings, or peer reputation in the technology community.

Table 6 - Ideal Characteristics of Experts [64]

- | |
|---|
| <ul style="list-style-type: none">● Generalists with a spread of interests and perceptions that give a high level of awareness in a broad context |
| <ul style="list-style-type: none">● Persons with particularly deep knowledge in a given field |
| <ul style="list-style-type: none">● Persons who are in or will be in positions which make it possible for them to affect the examined technology |

b. Methods of Collecting Expert Opinion

Individual Input

Individual input can be obtained in person, by telephone, mail, or by email. One-on-one interviews are either structured, focused, or non-structured. Structured interviews have an explicit set of closed questions. Focused interviews are directed to respondents who have pertinent knowledge of the topics to be discussed. The emphasis of this interview is to obtain subjective information regarding the situation under study. Non-structured interviews have no schedule or pre-specified set of questions, and respondents are encouraged to reveal their opinions and attitudes as they see fit. When using focused or non-structured interviews, overall objectives must be clear.

Committees, Seminars, and Conferences

These techniques may occur only when the experts are present in the same place at the same time. In terms of meeting frequency, committees meet most often, followed by

seminars and then by conferences. Committees are typically charged to specific tasks as opposed to seminars and conferences that are typically geared to information exchange. Expert opinion is usually collected from a committee of experts. This is because a collection of experts can provide a sum of information at least as great as that available to any single person. In addition, the number of factors considered is at least that of the total information available to a single person [70][81]. Martino provides the list of disadvantages with committees presented in Table 7. Much of this can be overcome with proper selection techniques, well-trained moderators, and proper expert selection.

Table 7 - Disadvantages of Committees [70]

- There is at least as much misinformation to a group as to an individual.
- Social pressure to agree with the majority can unduly influence the process.
- Reaching of agreement may become the goal.
- Repetition of arguments whether or not valid can influence the process.
- The process is vulnerable to dominant individuals.
- Group members may have a vested interest in certain points of view.
- The entire group may have a common bias.

Surveys

Surveys provide the most common means to solicit input from groups of experts when face-to-face meetings are impractical. This method is popular because it is quick, reasonably easy, inexpensive, and can be used to collect information from a large number of experts in a variety of formats [81]. Plus, it avoids the negative dynamics of face-to-face meetings. Unfortunately, traditional surveys are one-time affairs and do not provide

formal feedback loops. They do not directly address the handling of misinformation, but aim to cancel it out through averaging of responses. This technique also requires clear, concise, and unambiguous questions written with a common vocabulary. Survey interviews may be carried out face-to-face or through other forms of typed media or combinations, and they should be well constructed [36].

Delphi

Delphi provides a more interactive means of collecting expert opinion than its counterparts in expert opinion [63][70][78]. Originated in the early 1950s by the RAND Corporation, Delphi is designed to ensure a participant's anonymity, controlled feedback and iteration, and a statistical group response [81]. Delphi is composed of a series of questionnaires, or rounds, which are gathered, analyzed, and fed back to participants in the next round. Feedback includes the interquartile range of group response and rationale for disagreements and agreements with group feedback. Since participants generally act anonymously, individual tendencies to defend untenable positions are reduced, and the social demands to save face or retain credibility are reduced. Delphi has been extensively used in international studies to see upcoming events in the world of technology [19][49][70][81].

3. Trend Extrapolation

Much of the early work in the 1940s, 1950s, and 1960s revolved around trend

extrapolation as it attempted to map actual attributes over time [10][62][70]. It is important to identify the key factors considered as a proxy for the level of a technology. In the event that there are many, it is often recommended that a proxy index be considered to extrapolate the technology [69][81]. Additionally, the difference between a technology and a technical approach should be made [70]. A technology delivers an overall function to an end user, whereas a technical approach is merely a means to deliver that function. For example, aircraft engine technology provides thrust to an aircraft. This is accomplished by two technical approaches: turbo props and jet engines. Historically and in the future, other approaches have and may include rocket power, scram jets, or ram jets, all of which move the aircraft but by very different means. When examining hard disk drives, platter size indicates differing technical approaches for persistent storage. Technological trends tend to transcend those of technical approaches over time as limitations to current approaches are surpassed.

Christensen's discussions of disruptive hard disk drive technologies are in fact discussions of the technical approaches involved with platter sizes [26]. This provides an opportunity to introduce the concept of a technological envelope in Figure 3. Each technical approach, in this figure denoted as technology, is displaced by subsequent technologies to push the technological envelope forward. At its simplest, trend extrapolation is a basic model from which to derive future trends. Basic trend

extrapolation takes advantage of historical data and as such has been used as the basis for many technological forecasting studies and applications [9][10][62][70][81]. A common criticism of this method, however, is its inability to take into account outside influences and changing preferences [31][70][81][91][89][90].

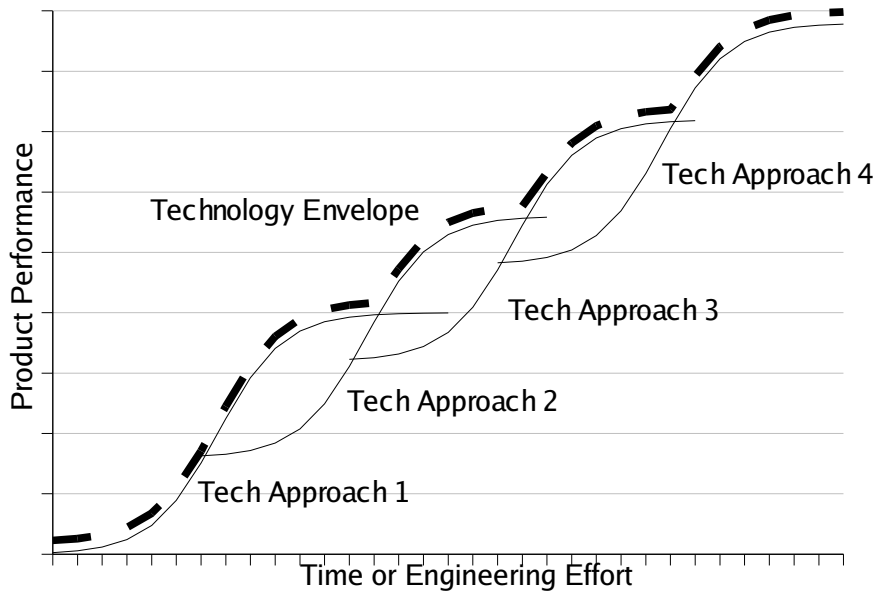


Figure 3 - Technology Envelope Built by Technological Approaches [26]

a. Naïve Models

In trend extrapolation, naïve models are basic, simple to use models. One of the most simple is the assumption that tomorrow will be exactly as today. This model is mathematically represented in (1) and is straightforward.

$$X_{t+1} = X_t \quad (1)$$

A more useful form is that of the constant growth models that assume the rate of

change will continue to be constant over time as represented by equation (2).

$$X_{t+1} = X_t + (X_t - X_{t-1}) \quad (2)$$

These basic models, although simple, provide guidelines for further development, the latter equation representing a limited adaptive means of evaluating future trends. Growth may also be represented as shown in (3), which is known as fractional or exponential [70][81].

$$X_{t+1} = X_t \left(\frac{X_t}{X_{t-1}} \right) \quad (3)$$

Perhaps the most famous fractional model is that proposed by Gordon Moore, co-founder of Intel and originator of Moore's Law in 1968.

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term, this rate can be expected to continue, if not increase. Over the long term the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be around 65,000 [73].

Although simple, Moore's Law has remained accurate over the last thirty years as displayed by Figure 4. The top trend line represents the number of transistors doubling every 18 months while the bottom trend line represents the doubling of transistors every 24 months. The additional plotted line is that of the actual number of transistors in products released by Intel. It is expanded in Chapter IV and [7].

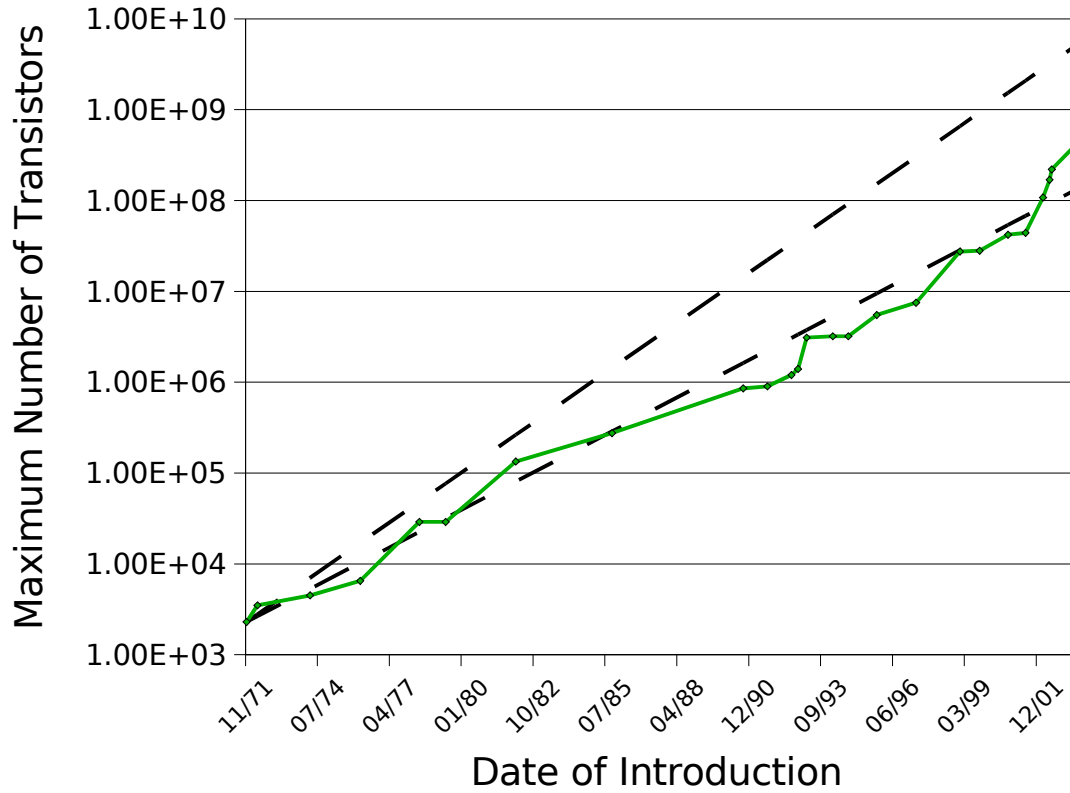


Figure 4 - Number of Transistors on Intel Microprocessors Versus Moore's Law [55].

b. Adaptive Weighting

Adaptive weighting provides a different approach of trend extrapolation. It attempts to smooth periodic fluctuations by taking into account a weighted sum of multiple time periods displayed in (4).

$$X_{t+1} = \sum_{i=0}^n w_{t-i} X_{t-i} \quad (4)$$

In (4), X_t represents variable X at time t , w_{t-i} represents the weight of variable X at time $(t - i)$ and n represents the number of time periods in the time window. The sum of

the weights is unity. This method can be further expanded to include a number of varying approaches to give strongest weights to the most recent observations. One common method is exponential smoothing where the weight is determined through (5).

$$w_{t-i} = \frac{a(1-a)^i}{d} \quad (5)$$

The constraints of (5) require $0 < a < 1$ and d to be the sum of the numerators for all w_i terms. This method can be used to provide the effect of a time window through using the last three periods. One can also extend this to put even more emphasis on more recent data [25].

c. Trend Analysis

Trend analysis provides further insight into technological forecasting. Using trend analysis in fairly stable environments in conjunction with expert opinion and monitoring provides much more insight. It is particularly useful when external mechanisms remain fairly constant. Predominant forms of trend extrapolation include exponential growth, growth curves, learning curves and substitution curves.

Exponential Curves

Exponential curves are particularly straightforward and tend to accurately model certain periods of technological development [25][51]. Over time, this rate will shift as a separate epoch is introduced. However, sociological and physical limits tend to slow or

stop the growth. The predominant means of accounting for these changing rates of change are S-shaped curves.

S-Shaped Growth Curves

S-shaped curves can be represented by a number of equations, typically derived from biological models that describe natural phenomena. They are characterized by slow early growth that steadily increases until a critical point of rapid growth. This growth then slows as the growth reaches a theoretical limit. Common models for this include the Pearl, Gompertz, and other logarithmic curves. These models are typically focused on the amount of market share held by a technology.

Over time, as technologies are replaced by others, many S-shaped curves are built to provide an envelope curve. Envelope curves are constructs of several sub-technologies used to forecast overall technological trends. One common example is presented by Martino, which tracks flight technology, measured by maximum speed, as it advances from piston engine biplanes to monoplanes to that of jet aircraft over time [70]. It can be said that the flight technology envelope continues to advance, despite the limitations of former leading technologies (piston aircraft), due to the advancement of previously unknown or limited technologies (jet aircraft). As the advancement of a certain technology stagnates, there may be replacements, assuming that demand for higher performance continues to grow. Most often, such trends are unidimensional and

correspond only to certain aspects within highly targeted segments.

d. Substitution Curves

Substitution curves are S-shaped curves which correspond to the rate that a technology or technical approach will replace its predecessor. In these cases, an older technology remains until the new technology has replaced it. In the beginning the inroads to market share are tough and slow, but as time progresses the advancement of the replacement technology overcomes that of the former technology until there is complete replacement, or a competitive equilibrium is achieved. Practitioners typically use those tools for S-shaped curves with the aim of mapping technology replacement as opposed to technology growth.

e. Linear and Multiple Linear Regression

Linear regression provides a means of taking past data and using them to estimate future values through simple linear models of the form (6). Although not always categorized to as extrapolative, Porter *et al.* classify them as such [81].

$$Y = a + b_1 X_1 + b_2 X_2 + \dots + \varepsilon \quad (6)$$

Where Y is the value to be predicted, a is the y intercept, b is the slope, and ε is the error which may be present. It is important to note regression does not in itself imply causality. By itself, regression merely indicates high correlation between the predicted value and those elements the value is regressed against. However, it may be used as a

tool from numerous perspectives to provide additional insight to the system being examined. Porter *et al.* provide four levels of regression from various standpoints [81].

1. Descriptive modeling – discovery of correlations, no basis for prediction
2. Simple prediction – causal process is not understood, data is assumed to make usable predictions
3. Causal models – understanding of the process and the causes of the future are known correctly; however, there is not a complete understanding of how they interact
4. Causal predictive models – causes and their effects are known, therefore it is used to further predict the current state

Ideally, all models are causal and predictive in that a complete understanding of the system is held and regression is used to quantify that relation. In practice, technology forecasting merely attempts to provide simple predictive models to determine what the capabilities of technologies are at some point in the future. Without an intrinsic understanding of a technology's functional and structural interrelationships, one has difficulty providing accurate models with these tools. Regression provides an educated guess as to what those interrelationships are however it assumes that the relationships are fixed.

4. Modeling and Simulation

When the relationships and impacts of environmental elements are known, modeling and simulation offer a rich tool set to forecast future technologies. Some commonly used tools include cross-impact analysis, Kane's Simulation (KSIM), system dynamics and

gaming. Although powerful, a major drawback of these tools is the need for deeper understanding of interrelations and factors involved in the overall technology. However, a major advantage is the power of these methods to provide new insights into the future.

Cross-impact analysis and time-based cross-impact analysis attempt to provide a future prediction with allowance to all interacting forces that shape that future [16]. From an initial state, events have probabilities of occurrence that are in turn affected by the occurrence of other events. In its most basic form, cross-impact analysis probabilities are static; however, time dependence can be modeled through Markov processes. Much of this approach was initiated from a game called “Future” by Gordon and Helmer for Kaiser Aluminum in the 1960s [52].

KSIM is a deterministic simulation model developed by Kane in 1972 that incorporates the idea of a cross-impact matrix to produce a relatively easy-to-use and powerful forecasting tool. Equation (7) summarizes the key mathematical relationship.

$$\frac{dX_i}{dt} = \sum_{j=1}^N \left(\alpha_{ij} + b_{ij} \frac{dX_j}{dt} \right) X_i \ln(X_i) \quad (7)$$

The X_i represents the i th impacted variable, N represents the number of variables, X_j represents the j th impacting variable, α_{ij} the “long term” impact of X_j on X_i , and b_{ij} the “short term” impact of X_j on X_i . The solution is shown in (8), where $P_i(t)$ is obtained through (9) [81].

$$X_i(t + \Delta t) = X_i(t)^{P_i(t)} \quad (8)$$

$$P_i(t) = \frac{1 + \frac{\Delta t}{2} \sum_{j=1}^N (|\alpha_{ij}| - \alpha_{ij}) x_j}{1 + \frac{\Delta t}{2} \sum_{j=1}^N (|\alpha_{ij}| + \alpha_{ij}) x_j} \quad (9)$$

KSIM utilizes a loose analogy to biological system growth. KSIM models can be constructed relatively quickly and easily. One major limitation is that complex interactions are limited to those between pairs of variables [69].

System dynamics (SD) looks for a more thorough understanding of the overall system expressed as a set of differential equations that are numerically integrated to derive behavior over time. The primary goal of the SD analyst is to understand how the complex web of feedback loops in the system leads to particular behavior or patterns [36]. SD models typically require that interrelations between system components be expressed as equations, making them difficult to implement for many researchers [81]. To apply SD to the process of technological change requires the forecaster to identify formulas describing interrelationships, which may not be readily apparent to the forecaster.

5. Scenarios

In technological forecasting, scenarios are internally consistent outlines of possible future worlds that can be readily applied to forecasting technology. Scenarios

incorporate several dimensions or important factors revolving around the technology being forecast. They typically incorporate uncertainty through a range of possibilities and are categorized as either future histories or future snapshots.

Future histories discuss events that may lead to an outcome. They are used for determining and assessing the path to a given state. The other type, a future snapshot, is a means of describing the future without regard to how one might get there. Both methods and combinations of the two provide insight into important issues and factors that may impact the future of a technology. Often, scenarios are the culmination of monitoring, extrapolation, expert opinion, and modeling, which bring to life the futures predicted by these other means. Since probability is involved, there are often pessimistic and optimistic versions.

6. Critical Review of General Technological Forecasting Literature

The information presented in the previous sections provides a general overview of the current methods available to the technology forecaster. Many of these methods rely on expert opinion and provide methods to determine future states of technology but often do not provide quantitative results. When it comes to quantitative forecasting of future technological capabilities, these methods have several outstanding research issues as Table 8 which have been gathered from the literature presented in the previous sections.

Table 8 - Research Issues in Technology Forecasting

Research Issue	Expert				
	Monitoring	Opinion	Extrapolation	Simulation	Scenarios
a: Accounting for characteristic interrelationships	yes	yes	no	yes	yes
b: Accounting for external influences	yes	yes	no	yes	yes
c: Accounting for changing preferences	yes	yes	no	yes	yes
d: Requiring historical data	no	no	yes	yes	no
e: Prone to expert error	yes	yes	yes	yes	yes
f: Adapting to disruptive technologies	yes	yes	no	yes	yes
g: Predicting future characteristics	no	yes	yes	yes	yes

a. Accounting for Characteristic Interrelationships

Understanding characteristic interrelationships for technology is very important.

Monitoring and expert opinion can specifically examine and account for any interrelationships which may occur since they are often point in time techniques.

However, when trying to extrapolate trends, aggregate variables may not properly take into account these interrelationships. When analyzing the advancement of technology, considering a single characteristic is apt to miss key elements such as those technologies which are not usually considered within the current environment but may displace the current ones.

b. Accounting for External Influences

External influences are prime drivers in the creation and growth of technology. If consumers and users did not demand higher levels of performance, those levels of

performance may never be obtained. Other issues that might influence the advancement of technology include scarcity of resources, or alternatives to the function of the technology. The whole point of monitoring and expert opinion is to determine the current state and future state of a technology. If a forecaster wishes to learn of external influences these may be built into the method. Simulation and scenarios can likewise address these issues, although they may require a very intimate knowledge of the environment. Traditional extrapolation techniques do not take this into account.

c. Accounting for Changing Preference

One thing that may result from external influences is changing preferences or trade-offs which face developers and researchers. This can be caused by a number of reasons, including obsolescence or new techniques to handle the same functions. Often times, as a technology matures the performance characteristics are no longer pushed to the limit because they have reached adequate levels to address the need of the product. In extrapolation, indices comprised of multiple attributes are constrained to constant trade-offs, but do not allow these trade-offs to change.

d. Requiring Historical Data

The requirement of historical data for forecasting has a number of positive and negative implications. Assuming past trends continue, one should be able to determine future capabilities and attributes of what a technology can attain. When provided with

the correct past data, there are many instances when future trends can be extrapolated while being less reliant on experts, or using them to a greater capacity to identify potential future events. Simulation and extrapolation use historical data to reinforce expert opinion or provide insight for further understanding of technological change. Monitoring, expert opinions, and scenarios do not need previous historical data, but are prone to expert error.

e. Prone to Expert Error

Although all methods are prone to expert error, those that rely primarily on experts have a larger risk. Scenarios and expert opinion rely on expert opinion, thus must be more careful when choosing experts. There are a number of techniques present to reduce the negative effects of dominant personalities and skewed panels, but in the end experts are still fallible. Extrapolation and simulation provide a means to augment the subjective evaluations of experts with better defined numerical results.

f. Adapting to Disruptive Technologies

Technologies which are out of the scope of the original forecast, or have capabilities that are not considered within the forecast can often become disruptive technologies. That is, unforeseen technology which replaces the current status quo. An ideal method would allow new technologies to be accounted for without much alteration to the current models. Since monitoring is an ongoing event, it can account for disruptive technologies

in that each time a technology is monitored, disruptive technologies can be identified. In similar fashion, expert opinion, simulation, and scenarios can also take into account disruptive technologies, assuming the experts are aware of what is going to occur. In those instances where the experts do not know, these methods will fall short.

g. Predicting Future Characteristics

Although this is the very definition of technological forecasting, monitoring does not provide this. In addition, when using aggregates or indexes, extrapolative techniques may not be able to predict these characteristics either. The other methods can often predict them as the experts provide the necessary insight.

h. Summary

Of the methods reviewed, there are a number of research issues which are present. Many of the issues revolve around technology trade-offs or characteristic independence and overall preferences. In effect, these shortcomings make it difficult to estimate future technologies because the definition of that technology is too rigid or imprecise to provide clear forecasts. Using single attributes or aggregates of attributes may not accurately represent the technology because most technology is a collection of components. By aggregating characteristics it at times becomes difficult to disaggregate any forecasts into the future characteristics of a technology. These issues can be addressed in part by expert opinion, scenarios, and simulation, but only when the interrelationships are

understood. If there are changes to these interrelationships or external influences not foreseen by the forecaster or their experts, the applicability of these methods are more prone to disruptive technologies.

Often times technologies which are outside the scope of the forecasts have the greatest impact on the future of a technology. These disruptive technologies are often alternative approaches to providing similar functions. Because their function is similar to that of the technologies being forecast, they may be a viable alternative to the approaches being studied. However, the criteria by which they are judged is different; and as such may not be excluded from the analysis. Organizations that depend on their experts and customers to tell them what would be needed in the future, often effectively disregard the approaches that can put them out of business. For quantitative analysis, extrapolation is the best technique but fails to take into account many outstanding issues revolving around multiple characteristics. To address these issues, one approach that technology forecasters employ is that of state-of-the-art and trade-off surfaces.

B. State-of-the-Art and Trade-off Surfaces Literature

To overcome some of these weaknesses, it is useful to view the SOA for a given technology as a combination of indicators. For the purpose of this study, the SOA is defined in Dodson's work as the "state of best implemented technology as reflected by the physical and performance characteristics actually achieved during the time period in

question [31].” Therefore, SOA represents the best level of technology that is readily produced.

1. General Procedure

The current SOA is determined by analyzing products that are instances of the technology under study. In addition, to understand the efficiency of that technology, there are structural elements that must be taken into account [2]. Shifting preference functions or trade-offs should also be considered. At any point in time, SOA may be represented by a broad array of functional requirements and structural components [9] [67]. In aircraft, for example, one common trade-off is the amount of speed vs. cargo capacity. It should be noted that the concept of SOA surfaces is a direct measure of technology and as such, typically takes into account factors such as economic feasibility, need, and value by assuming fixed relations. To measure the SOA, the following steps must be performed [46].

1. Specifying an operational definition of SOA
2. Specifying the general form of the SOA surface
3. Developing a technique to represent data as a surface
4. Specifying the guidelines for parameter selection of the SOA
5. Develop prospective SOA advances

Dodson formalized the concept of SOA surfaces, stating that methods implementing

SOA often lack the precision required for effective analysis [31]. A generalized approach to quantitatively assess the SOA uses structural and performance characteristics of products. These characteristics are used to represent a multidimensional SOA surface which can be monitored over time to track technological change [2][31].

A simple example of a SOA frontier is presented in Figure 5 indicating increasing distance from the origin of a surface [31][32][67]. The SOA represents a trade-off surface where individuals may choose to have more of one characteristic than another. Once the SOA surface is mapped, it is possible to extrapolate the movement of that surface over time. Prior data is used to project a surface of an ideal form, which can be used to determine the future SOA. Since this surface consists of a number of physical and performance attributes at a given point in time, formulating an accurate means to account for trade-offs and technological progress becomes the challenge. To address the issue of incommensurate units, ratios can be used.

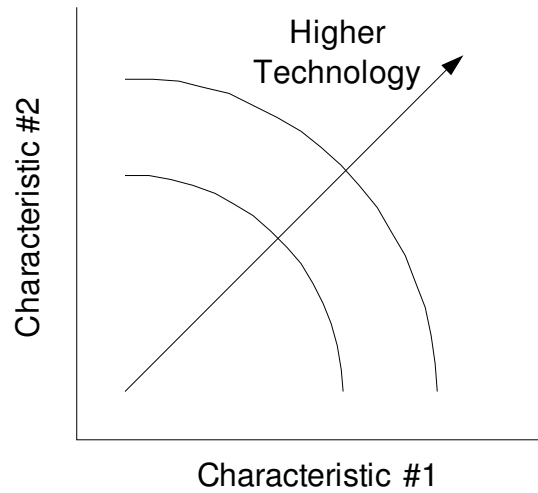


Figure 5 - Advancing Technology Frontier

There are many approaches that address technology trade-offs in the form of technological indices. These methods take into account the functional performance of a technology, as defined by user requirements. Technical parameters measuring structural characteristics necessary to perform the function being delivered by the technology are also taken into account [1]. For example, some functional characteristics of transportation technology would be speed and cargo capacity while energy consumed and cost could be structural. By taking into account both structural and functional characteristics, it is easier to more accurately assess the level of technical accomplishment [2].

2. Scoring and Regression

Scoring and regression are two common means to measure the SOA. Using

methodologies that are similar, both economists and technologists have worked to forecast technology [104]. The major difference between their approaches is the value that is forecast [2]. In general, economists believe that technological change provides a constant level of performance for an ever decreasing cost, while technologists focus on the increasing functionality of technology through indices.

Hedonics, the economists' approach developed by Court and later Lancaster and Griliches, is based on the hypothesis that a product's utility is a function of its essential characteristics such as size, power, comfort, and fuel economy in the case of automobiles [30][47]. Product price variation is explained by the differences in these characteristics. Over time a given set of characteristics and their values experience price changes that are decomposed into a "quality" or a "technological change" effect and a "pure price" effect. Pure price changes are associated with the changing value of a characteristic set, while technological change may be thought of as the difference between actual price changes and pure price changes. A major limitation of this model is the assumption that products with fixed functionality reduce in price over time, as opposed to a product that experiences decreasing or increasing functionality. In other words, economists tend to look at economic factors when determining a trade-off surface when they should be examining technical factors [104].

Technologists proposed an operationally similar technique known as the RAND

technique, which was based on Dodson’s earlier work [2][31]. This technique regresses data on characteristics relating to the principal axis of time, as opposed to the economists’ target of price. The advantage of the RAND technique is that it doesn't utilize economic variables in the measurement of technological change and thus bypasses many of the limitations of the Hedonic method [92]. However, it shares many of the same drawbacks [92]. Primarily, the shape and value of trade-offs on the SOA surface is assumed constant over time. In this, both methods are tied to particular non-flexible application spaces or markets. Additionally, key elements may be correlated, resulting in inaccurate or flawed regression models. A suggested resolution for this is the use of Principal Component Analysis to merge correlated attributes into proxy attributes [93].

Gordon proposes a version of this approach that maps SOA characteristics with a scoring model [46]. Later, this was assumed to take the shape of an S-curve over time [46]. By implementing precise measures via dimensionless technology indexes, forecasters can get a better “picture” of the technology [35]. Gordon’s model is based on Knight’s prior work in the forecasting of technology change in the computer industry [60]. Knight’s work determined the advancement of computer technology through (10).

$$SOA = K_1 \left(\frac{P_1}{P'_1} \right) + \dots + K_n \left(\frac{P_n}{P'_n} \right) \quad (10)$$

Gordon added the concept of a primary parameter, P_1 , without which the technology

would not be considered as an SOA candidate. Gordon and Munson's model is given in (11) [46].

$$SOA = \frac{P_1}{P'_1} \left[K_2 \left(\frac{P_2}{P'_2} \right) + \dots + K_n \left(\frac{P_n}{P'_n} \right) \right] \quad (11)$$

In (11), n represents the number of parameters in the analysis, P_n represents the value of the n^{th} parameter, P'_n represents the reference value of parameter P_n , and K_n represents the normalized weight or trade-offs of parameter n . In this equation, if P_j is 0 then it can easily be deduced that the product will not achieve the SOA.

Determination of the weights used in (10) and (11) may be done through expert opinion or statistical techniques, such as multiple linear regression and factor analysis [35][46]. A common illustration of these methods is the examination of aircraft SOA [1] [31][67]. A major disadvantage with this approach is again that it assumes constant trade-offs over time [59][90].

3. Dodson's Approach

In his original work [31][67], Dodson proposed that technological progress could be mapped to the movement of SOA surfaces represented by a hyper plane or ellipsoid surface as denoted in (12) and (13) where X_i represents the i^{th} variable and α_i represents the i^{th} parameter or zero intercept along the i^{th} access.

$$\frac{X_1}{\phi_1} + \frac{X_2}{\phi_2} + \dots + \frac{X_n}{\phi_n} = 1 \quad (12)$$

$$\frac{X_1^2}{\alpha_1^2} + \frac{X_2^2}{\alpha_2^2} + \dots + \frac{X_n^2}{\alpha_n^2} = 1 \quad (13)$$

One of the major difficulties of fitting this or any other surface is the ability to determine the best fitting surface. In his work, Dodson proposes an ellipse based on a least squares fit [31].

Dodson further postulates that advancement of technology can be represented by (14) where γ denotes the advancement, ϕ_n represents the radial distance to an observation point beyond the SOA, and ϕ_{SOA} represents the radial intercept of the SOA.

$$\gamma = \frac{\phi_n}{\phi_{SOA}} \quad (14)$$

Martino confirms the validity of Dodson's work when he notes that different types of aircraft, transport or fighter, may share the same trade-off surface [67]. Further work indicated that the parameters characterizing a given technology are divided into two subcategories: performance and technical parameters [91]. Although at times these trade-offs are somewhat constant, they are likely to shift and change over time [90].

Criticisms of this approach include the fact that the ellipsoid surface only has a small number of observations on its surface, and it is subject to influence from extreme outliers [68]. Additionally, like regression, it fails to take into account the correlations between

the various characteristics, which may significantly distort the overall rate and extent of technical progress [89]. In addition, it is difficult to compute the theoretical surface, which was further supported by Dodson's later adoption of the RAND method [32].

4. Technometrics

Although lauded as a worthy approach and philosophy, Dodson's work has some implementation problems as mentioned above. Its elegant means of handling trade-offs was thought of as a good step forward, but difficult to achieve. Sahal introduces Technometrics to address these issues through discriminant analysis [89][90]. In his work, Sahal proposes a pair of technology indices, design and performance characteristics, mapped to trade-off possibility curves over time. Design and performance characteristics are combined to form dimensionless technology measures. These non-dimensional measures can then be used in conjunction with discriminant analysis to determine proper groupings of technologies. The index then takes the form presented in (15).

$$y = a_1 x_1 + a_2 x_2 + \dots + a_p x_p \quad (15)$$

In contrast to the forms presented by RAND and Hedonics, x_i represents the i th dimensionless technology measure that is derived from product characteristics. The coefficients, represented by a_i , are generated by maximizing the ratio of the *between group* variance and *within group* variance using discriminant analysis. This creates a

fixed equation for each grouping of technology to determine a technology score [91][89]. Sahal points out the majority of technological changes occur at the level of the features rather than the underlying phenomena. This method offers the advantage of mapping technological change within like groups and allowing the function to change as time passes. It also does not suffer some of the drawbacks of pure Euclidean distance as it aims to maximize the distances while taking into account their statistical distribution. Disadvantages with this approach are that the transformation of data to dimensionless values is in itself a discriminant function; it requires more data, and it is computationally intense.

5. Critical Review of SOA and Trade-off Surface Literature

SOA and trade-off surfaces provide a method to combine the benefits of monitoring, expert opinion, and trend extrapolation to provide a more thorough view of technological progress. Although this is an improvement over other extrapolative techniques, Table 9 lists a number of outstanding research issues presented earlier and is explained in the subsequent text.

Table 9 - Current Issues with SOA Literature

Issue	Techno-			
	Dodson Scoring Regression metrics			
a:Emphasizing best available technologies[31]	no	no	no	no
b:Allowing trade-off change[9][37][91][104]	yes	no	no	yes
c:Allowing new/old characteristics[2][104]	no	yes	no	yes
d:Predicting future characteristics[68]	no	no	yes	no
e:Handling correlated characteristics[2][37]	no	no	no	no
f:Handling multiple outputs	no	no	no	yes
g:Handling variable returns[65]	no	no	no	yes

a. Emphasizing the Best Available Technologies

E. N. Dodson suggests that the SOA represents the best available technologies at a given point of time [14]. With new product development, or acquisition of future technologies, a person or organization may wish to know whether or not a technology will be feasible. In Christensen's *Innovator's Dilemma*, disruptive technologies are often those which are out of the current scope of a forecast, pushing envelopes beyond an average composite of the current industry [26]. Emphasizing the best technologies provides a means to measure the effective limitations of future technologies.

Additionally, products are released that are not SOA, meaning that all products should not set the future targets for technological limits. Dodson's technique, scoring models, regression, and technometrics do not base their representation of the SOA with the best technologies, but use averages.

b. Allowing Trade-off Change

Similar technology can be used in multiple markets. Because of this, product designers are faced with a number of trade-off decisions. Choosing one performance characteristic or attribute over another may or may not indicate a different level of technology and as such any representation should take this into account. By allowing these multiple sets of trade-offs, it is possible to represent multiple technical approaches on the same surface. By building an elliptical surface, Dodson provides a means to

represent changing preferences; although the method does not well represent the best technologies [31]. Sahal and his technometrics provides for this by clustering like technologies together [91]. Scoring and regression models however do not permit this tendency because they set fixed weights. These fixed weights, once set affect all future forecast and assume a constant preference function throughout time. This carries the implication that once a technology is chosen, the preference structure for measuring a technology is never altered.

c. Allowing New/Old Characteristics

Technology's evolution over time has potential to introduce and retire product characteristics and attributes. Early characteristics or attributes may become irrelevant as time passes. This is the case when a certain level of achievement has been attained that there is little reason or motivation for the attributes trend to continue. One common example is size. Many technologies aim to reduce the size of products, however at some time reduction in size may or may not diminish the purpose of the products in question. On the opposite end of the scale, it is not uncommon for new attributes to become a driving force in technological advancement. This may be the case if former attributes are no longer relevant, or if additional functionality has been added to the products. Scoring models and technometrics allows for this, but the others do not.

d. Predicting Future Characteristics

The purpose of forecasting is to determine the future characteristics of a useful machine [70]. In extrapolative and other forecasting, this means a set of future characteristics should be determined through the methodology. Interestingly, although technometrics and scoring provide a means to monitor and assess technology and its progress, they fall short on providing a means to determine future characteristics of the products they analyze. Scoring does not allow disaggregation of the trade-offs to determine the future characteristics of a product [68]. Technometrics, although suggesting a possible means to assess change, does not actually provide a means to do so.

e. Handling Correlated Attributes

Another issue faced by the forecasting of technology is that of correlated attributes. Often times it is very difficult to isolate characteristics and attributes from each other; also, as technology changes, previously correlated characteristics may be decoupled over time. This becomes a difficult issue for the technology forecaster to address. Dodson, Sahal, and regression techniques all rely on attribute independence and lack of correlation between variables. While factor analysis can be used to convert correlated attributes into uncorrelated independent variables that can be used in regression, this sacrifices the direct interpretation of the attributes and prediction of future characteristics. In addition, scoring methods may be unduly influenced by correlated

variables as the the compounded attributes may unduly effect technology scores.

f. Handling Multiple Outputs

It is important to take into account a number of performance attributes with technology forecasting. Because a product or machine may provides several functions, or a function may be composed of several characteristics, there is a need to take them into account. With the exception of technometrics, most current methods do not address the concept of multiple outputs. In the case of scoring, attributes may be presented in the formulation, however it does not allow for disaggregation of capabilities, and the score is considered absolute. Multiple outputs could be accommodated through the use of either response surface methods or treating them as additional independent variables. The former creates additional methodological complexity and has not been used in practice for technology forecasting. The latter likely exacerbates the issue of correlation among attributes.

g. Handling Variable Returns

While the concept of returns to scale is well studied and understood in the field of economics, existing technology forecasting techniques do not allow for the flexibility of different returns to scale models. To illustrate this and relate it to the economic concept of returns to scale, think of price vs. performance for a computer system. At the low price segment, small budget increases would typically yield sizable performance

increases. Numerically, a 10% budget increase might result in a 20% performance improvement. This corresponds to a product that is in the increasing returns to scale region. In contrast, a computer that is in the high price range would typically require a large spending increase in order to make a small performance increase, say 20% increase in cost to get only a 5% increase in performance. This computer would be then characterized as being in the decreasing returns to scale region. Although models could be built using increasing or decreasing returns to scale, a comprehensive and flexible technology forecasting technique should be able to allow for both simultaneously to even model a simple case such as the computer price-performance case. Allowing for both simultaneously is referred to as variable returns to scale.

h. Summary

SOA and trade-off surfaces were generated to address the research issues with extrapolative technology forecasting as presented in Table 8. These surfaces were meant to provide tools for forecasters to represent the technological state of the art and use it to forecast future trends. The overview of the literature presented in this work reveals a number of outstanding research issues that are not addressed by these techniques.

Although work in the subject of technological progress has been extensive over the last half century, the fundamental issues are presented in Table 9 are shown in Figure 6.

In this figure, Dodson addresses the monitoring of technology using fixed trade-offs.

		Category	
		Monitoring	Extrapolation
Weighting	Fixed	Dodson [31][67][70]	Scoring and Regression [1][2][31][67] [88][104]
	Variable	Sahal Techonometrics [89-92]	Unaddressed

Figure 6 - Techniques to Model the SOA

This provides the forecaster a method to assess the current levels of technology but not a way to extrapolate them into the future. Scoring provides a means to forecast the state of future technology but does not allow the forecaster to determine the characteristics of the technology being forecasted. Regression models provide a way to extrapolate future trends and characteristics of a technology, but do not address the dynamic nature of those trade-offs and assume that past relationships will remain constant. Other shortcomings of regression and scoring is their inability to handle correlated data. All of the methods do not measure the best available technologies, they measure approximations of the mean or central tendencies, and those methods that do extrapolate the future characteristics of a

technology, do not allow for the trade-offs and preferences of those characteristics to change.

C. Data Envelopment Analysis Literature

First introduced in 1978 by Charnes, Cooper, and Rhodes [21], DEA has been used in more than 1500 articles over the last two decades to measure productivity in applications [18]. Unlike central tendency approaches such as regression, DEA is an extreme point technique that seeks the best performance of the data set that is evaluated. Each dataset is composed of a number of decision making units (DMUs). DMUs are entities evaluated as being responsible for converting inputs into outputs [18].

1. Introduction

DEA is an extension of Farrell's [43] relative efficiency, θ , presented in (16).

$$\theta = \frac{\sum \mu_r y_r}{\sum v_i x_i} \quad (16)$$

In this equation, μ_r represents the weight of the r th output, y_r , and v_i represents the weights of the i th input, x_i . The CCR ratio form (17) is an expansion of (16).

$$\begin{aligned}
\max \quad & \theta_k = \frac{\sum \mu_r y_{r,k}}{\sum v_i x_{i,k}} \\
\text{s.t.} \quad & \frac{\sum_{r=1}^s \mu_r y_{r,j}}{\sum_{i=1}^m v_i x_{i,j}} \leq 1 \quad \forall j \in \{1 \dots n\} \\
& \mu_r, v_i > 0.
\end{aligned} \tag{17}$$

In this set of equations, θ_k represents the relative efficiency of DMU k , n represents the number of DMUs, s is the number of outputs, and m is the number of inputs. Formulation (17) can be interpreted from a self-evaluation perspective as DMU k selecting ways to value good attributes (y) and bad attributes (x) to achieve as high a score as possible. There are only two constraints on the weights that can be used. First, no DMU can get a score better than 1.0 using the weights selected by DMU k . Second, the weights must be positive.

This can be further transformed to a linear program as described in (18), which is known as the DEA multiplier input-oriented model. The program is solved once for each DMU to determine their efficiency scores.

$$\begin{aligned}
\max \quad & \theta_k = \sum_r \mu_r y_{r,k} \\
\text{s.t.} \quad & \sum_i v_i x_{i,k} = 1, \\
& \sum_r \mu_r y_{r,j} - \sum_i v_i x_{i,j} \leq 0, \quad \forall j \in \{1 \dots n\} \\
& v_i, \mu_r \geq 0.
\end{aligned} \tag{18}$$

Another general form of DEA is the envelopment model, which constructs virtual DMUs from a weighted combination of its peers. For a more detailed analysis of the relationship between the multiplier and envelopment models one is referred to [28][29]. The input-oriented envelopment model is described in (19). The variables $x_{i,k}$ and $y_{r,k}$, represent input i and output r of DMU k . Instead of the input and output weight variables (v and y), from the multiplier model, the envelopment model uses λ and θ .

$$\begin{aligned}
 \min \quad & \theta_k \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{i,j} \lambda_j \leq \theta_k x_{i,k}, \quad \forall i \in \{1 \dots m\} \\
 & \sum_{j=1}^n y_{r,j} \lambda_j \geq y_{r,k}, \quad \forall r \in \{1 \dots s\} \\
 & \lambda_j \geq 0, \\
 & \theta_k \geq 0.
 \end{aligned} \tag{19}$$

The variable, λ_j , represents the weight of DMU j used to set the performance target for DMU k which is being evaluated. The left hand sides of the two constraints in (19) can be thought of as setting a performance target for DMU k based on the best possible combination of the n DMUs. The first summation states that any input constructed of other DMUs will have the same as or less input than the evaluated DMU k . The second constraint sets the summation of the outputs of the weighted combination of outputs to at least the output of DMU k by setting its output to at least $y_{r,k}$. In the event that $\theta_k < 1$,

DMU k is using too much input to deliver the output relative to its peers. When $\theta_k = 1$, no DMU can produce as much output as DMU k with less input, thus meaning that DMU k is efficient.

Another way to approach the same problem is to use that of output-orientation as represented in (20).

$$\begin{aligned}
 \max \quad & \phi_k \\
 & \sum_{j=1}^n x_{i,j} \lambda_j \leq x_{i,k}, \quad \forall i \in \{1, \dots, m\} \\
 & \sum_{j=1}^n y_{r,j} \lambda_j \geq \phi y_{r,k}, \quad \forall r \in \{1, \dots, s\} \\
 & \lambda \geq 0.
 \end{aligned} \tag{ 20 }$$

The objective function ϕ_k represents the output-oriented relative efficiency of DMU k relative to the other DMUs. The summations are similar, however, in cases where $\phi = 1$, a DMU is considered efficient and unable to deliver any more output for its input based on what other DMUs produce. In the event that $\phi_k > 1$, DMU k would be expected to deliver more output than it is currently doing based on its peers.

One fundamental assumption of the efficiency envelope is that any convex combination of efficient DMUs is feasible. This envelope provides a benchmark for those DMUs that are not efficient. Since DEA does not assume any specific underlying functional form, each DMU has a degree of freedom in selecting the importance of each

input and output. By allowing for flexible trade-offs, DEA maximizes DMU efficiency relative to other DMUs by exploiting each DMU's strengths.

2. Weight Restrictions

Because DEA offers complete flexibility in weighting of inputs and outputs, its results can at times prove to be inconsistent with prior knowledge [3][115]. Weight restrictions may be categorized into three types:

1. Direct weight restrictions.
2. Adjusting observed input/output levels to capture value judgments.
3. Restricting the virtual inputs and outputs.

Direct weight restrictions are performed by introducing restrictions within the linear program to set relative ordering of overall weights, assigning relationships to the inputs and the outputs, or forcing minimums or maximums to guarantee elements be included.

Adjusting observed input/output levels may be done by transforming data to correspond with *a priori* weight restrictions prior to analysis [22], or by setting ordinal restraints for DEA weights in the multiplier model [44]. Restricting virtual inputs and outputs are done with additional constraints of the ratio of outputs to input [107]. A general overview of the procedures can be obtained in [3].

3. Malmquist Productivity Indices

The Malmquist productivity index (MPI) evaluates efficiency changes over time as an

extension of the concept initially introduced by Malmquist and Shepard [66][98]. The application was expanded and applied to DEA [41][42][48]. The MPI is measured as a product of the relative movement of a DMU with respect to its peers, the catch-up effect, and the relative movement of the efficiency frontier, or frontier shift.

Since the MPI analyzes the DMUs changes in relative efficiency over time, additional naming conventions are required. The inputs and outputs for a DMU k at a given time t is represented by $(x_k, y_k)^t$. At the time $(t+1)$, the inputs and outputs for the same DMU k are represented by $(x_k, y_k)^{t+1}$ represents the inputs and outputs at time $t+1$. Mathematically the catch up effect, or relative movement of DMU k to its peers is displayed in (21).

$$C_k = \frac{\delta^{t+1}(x_k^{(t+1)}, y_k^{(t+1)})}{\delta^t(x_k^t, y_k^t)} \quad (21)$$

The variable C_k represents the catch-up effect of DMU k while δ^t and δ^{t+1} can be thought of as the efficiency with respect to all of the DMUs at time t and $(t+1)$. This value, C_k , is very similar to Dodson's γ for measuring the change of technology indices over time. The frontier shift is represented by (22).

$$F_k = \left[\frac{\delta^t(x_k^t, y_k^t)}{\delta^{t+1}(x_k^{(t+1)}, y_k^{(t+1)})} \times \frac{\delta^t(x_k^{(t+1)}, y_k^{(t+1)})}{\delta^{t+1}(x_k^{(t+1)}, y_k^{(t+1)})} \right]^{\frac{1}{2}} \quad (22)$$

Multiplying C_k and F_k provides the MPI for a DMU at a given time period as displayed in (23).

$$MPI_k = \left[\frac{\delta^t(x_k^{(t+1)}, y_k^{(t+1)})}{\delta^t(x_k^t, y_k^t)} \times \frac{\delta^{t+1}(x_k^{(t+1)}, y_k^{(t+1)})}{\delta^{t+1}(x_k^t, y_k^t)} \right]^{\frac{1}{2}} \quad (23)$$

The MPI was designed to address individual DMUs whose performance changes over time, which is generally not the case with many technology product offerings. This is due to the fact that many products are offered once and are in turn replaced with different products, rather than improving a product once it is purchased. In instances where a DMU does not change from time period to time period, the inputs and outputs remain the same which can be expressed in (24).

$$(x_k^t, y_k^t) = (x_k^{(t+1)}, y_k^{(t+1)}) \quad (24)$$

Combining (24) and (23) further reduces MPI_k to 1. In the event of products that are released only once, the MPI does permit the tracking of technological progress over time.

For further discussion of MPI and like indices, one is referred to [40][39][42].

4. Strengths and Limitations of DEA

DEA is a powerful tool to assess relative efficiency of multiple DMUs. It has many strengths presented in Table 10 [6].

Table 10 - Strengths and Limitations of DEA

<i>Strengths</i>	<i>Limitations</i>
<ul style="list-style-type: none"> • DEA can handle multiple inputs and outputs simultaneously 	<ul style="list-style-type: none"> • Since it is an extreme point method, noise can cause problems.

<i>Strengths</i>	<i>Limitations</i>
<ul style="list-style-type: none"> DMUs are independently rated against peers which are identified explicitly by DEA 	<ul style="list-style-type: none"> It provides only relative efficiency, not an absolute efficiency.
<ul style="list-style-type: none"> DMU characteristics may have heterogeneous units 	<ul style="list-style-type: none"> It can be computationally intensive
<ul style="list-style-type: none"> There is no <i>a-priori</i> weighting of attributes relating inputs to outputs 	<ul style="list-style-type: none"> Statistical hypothesis tests are difficult

5. Procedure for Implementing DEA

Due to the nature of DEA, it is important to follow a set procedure prior to implementation. The procedure for implementing DEA based on Anderson [6] is summarized in Figure 7, and explained in this section. This model will be later expanded

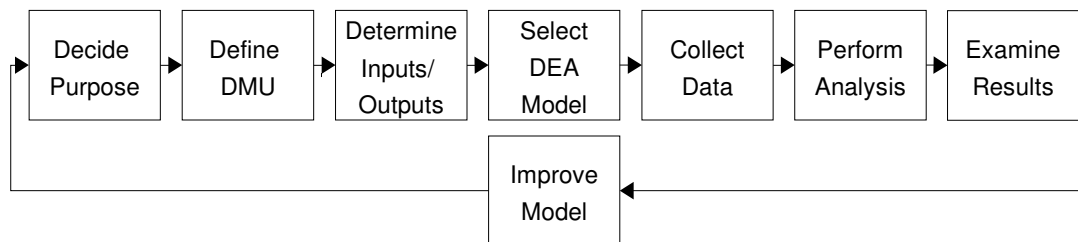


Figure 7 - Procedure for Implementing DEA [6]

to forecasting of the SOA.

a. Decide Purpose

Before conducting any analysis, it is important to determine its purpose. Defining the purpose permits accurate identification of required data. In technological forecasting, the purpose is to determine the “state” of a technology.

b. Define DMU

The determination of the characteristics to be studied can be difficult. It is important to select characteristics that are shared amongst all DMUs being evaluated. A general guideline for DEA is the number of DMUs should be three times the sum of the number of inputs and outputs [6][12][23].

c. Determine Inputs/Outputs

Once a DMU is defined, its input/output model must be constructed. This process can be very involved and require intimate knowledge of the subject matter. Since the goal of productivity analysis is to maximize a set of outputs from a minimized set of inputs, one method of characterization is to associate “bad” characteristics with inputs and “good” characteristics with outputs. This provides a means to assess efficiency when there may not be a clear-cut definition of what is being produced.

The purpose of the study can provide a starting point for output classification. Outputs correspond to that which a DMU is responsible for producing. Common inputs include the man-hours and capital resources used by the factory to produce the products which are considered outputs. In the event that increasing values of an input or decreasing an output are considered beneficial, mathematical transformations may be required. Basic transformations may include inversion or subtraction from a theoretical maximum.

d. Select DEA Model

There are a number of models to choose from in DEA. One major decision discussed earlier includes models using input-orientation vs. models using output-orientation.

Another consideration to make is that of returns-to-scale (RTS).

Input Versus Output-Orientated Models

One decision to make when selecting DEA models is that of orientation. The difference can be best understood by examining the graphical example in Figure 8. One considers whether or not the goal of the DMU is to minimize the input for a given output,

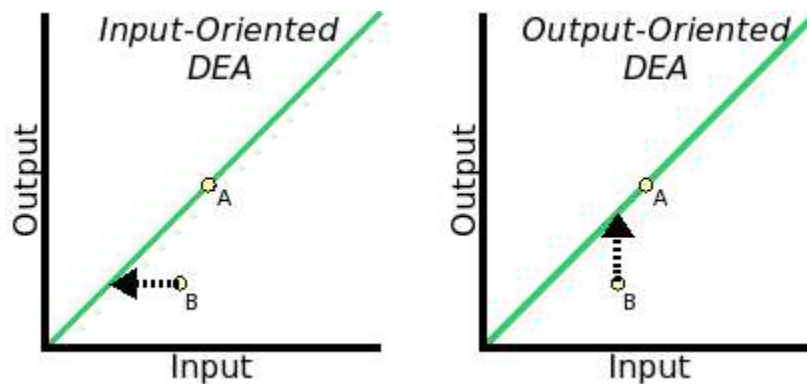


Figure 8 - Graphic Representation of Input vs. Output-Oriented

an *input-oriented model* (IO), or maximize the output for a given input, an *output-oriented model* (OO). An efficiency score of 1.0 indicates an efficient observation for both methods. With IO models, efficiency scores less than 1.0 indicate the same amount of output can be achieved with less input. Conversely, with OO models, efficiency scores greater than 1.0 to indicate that more output can be achieved with the given input.

Returns to Scale

An important issue to consider are the returns to scale for DEA. Returns to scale refers to changes in efficiency based on size. Straightforward, two-dimensional examples of returns to scale are illustrated in Figure 9. Constant returns to scale (CRS)

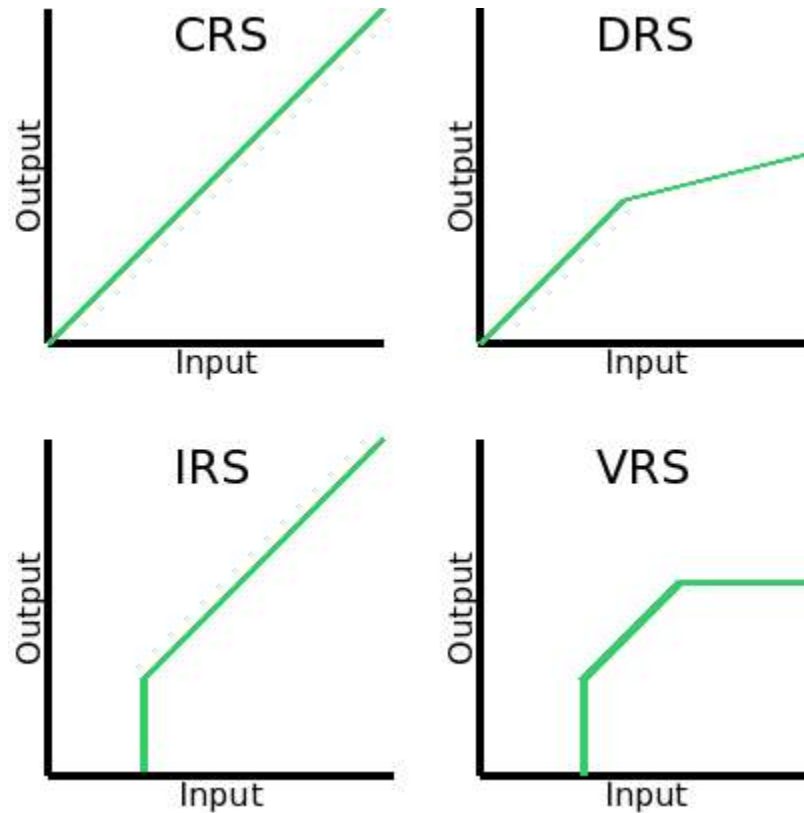


Figure 9 - Graphical Representations of Returns to Scale

implies that production is linearly scalable regardless of the amount of inputs or outputs involved. This can be mapped to a constant return from zero to infinite inputs. This is a large assumption and typically holds true for only a small subset of values.

Increasing returns to scale (IRS) refers to increasing returns as the amount of outputs and inputs increase. Using the same example of computers, the cost of entry may reflect

a minimum performance cost ratio that is significantly higher than improving the performance from that base cost of entry. When using IRS, those items that may not be efficient in CRS may be efficient. Decreasing returns to scale (DRS) refers to diminishing returns for additional input beyond points of inflection.

One common example is that of computing performance. Although somewhat scalable, getting a five percent increase in performance from the best performing computers may cost significantly more than a five percent increase for the rest. By utilizing a DRS model, it is possible for DMUs that would not have been efficient using CRS to be efficient because it may not be possible get the same returns to scale to infinity.

In DEA, the combination of these approaches is termed variable returns to scale (VRS). This model provides the benefit of both IRS and DRS by taking into account the minimum cost of entry and the diminishing returns which many technologies may experience. Mathematically, returns to scale is performed by adding the constraints to the appropriate envelopment model as described in Table 11.

Table 11 - Common DEA Models Available

<i>Returns to Scale</i>	<i>Constraint</i>
Constant	$\lambda \geq 0$
Increasing	$\sum \lambda \geq 1$
Decreasing	$\sum \lambda \leq 1$

<i>Returns to Scale</i>	<i>Constraint</i>
Variable	$\sum \lambda = 1$

A more comprehensive treatment of DEA is available in [18] and [23].

e. Collect Data

Once the model is chosen and defined, the data must be collected. If there are high correlations between inputs or between outputs, then the need for data points may be reduced [6].

f. Perform Analysis

There are numerous means to solve the linear programs required for DEA including mathematical programming languages AMPL, LINDO, LINGO, and LPSOLV. In addition to these are [13][77][106] and [114].

g. Examine Results

Because DEA is an extreme point technique, a single outlier can have a major impact on results. For the SOA, this provides the benefit of setting performance benchmarks to the best available technologies. However, in the event of unrealistic weighting, it may be necessary to implement weight restrictions. Examination of those DMUs comprising the efficiency frontier may provide insight into the need for weight restrictions. Examination of inefficient DMUs may reveal that there are characteristics that are not being measured and thus should be added. If there is little differentiation between efficient and non-

efficient DMUs this may indicate that some important attribute is not being considered.

Additional considerations should be given to sensitivity of results as provided by Cooper *et al.* [18][96][113].

h. Improve Model

Based on the information provided in the prior steps, a practitioner of DEA may choose to improve the model being used and repeat the above steps. In the event where there are many attributes, a practitioner may want to remove inputs and outputs to examine the overall sensitivity of the system. If there are dominant DMUs, such as those with large λ 's for many DMUs, it may be possible to identify key attributes which are or are not important. Some improvements may include the elimination of some inputs that are not necessary for the model or those characteristics that do have small or large effects on the overall efficiency.

6. Example of Using DEA to Evaluate Technology

Now that the framework for use of DEA has been presented, an example will be provided following the procedure presented in Figure 7 by applying the technique to a simple two dimension example, USB pen drives.

a. Decide Purpose

The purpose of this study is to evaluate the level of technology for USB pen drive technology.

b. Define DMU

Since flash memory can take many forms, and we are interested in the function of highly portable USB storage regardless of the underlying technical attributes (type of memory, size of silicon, among other elements) a DMU is to be considered a USB pen drive.

c. Determine Inputs/Outputs

USB pen drives provide end-users with the ability to store data in a portable format replacing much of the former purposes for floppy disk drives. As the function of the pen drive is storage, overall capacity will be considered as the output required. The second differentiating factor between USB pen drives is that of cost, which will be our input.

This model is reflected in Figure 10.

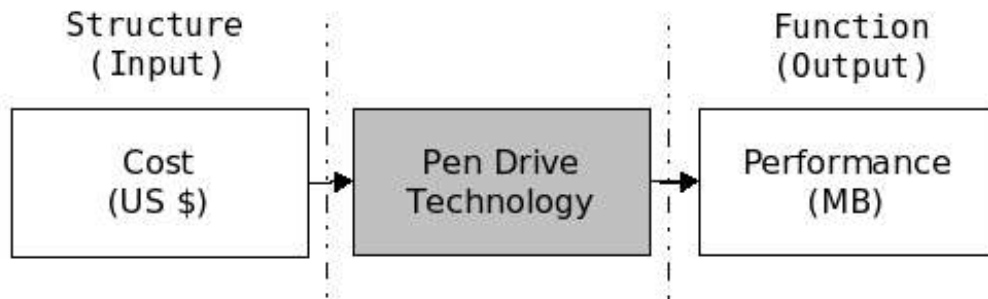


Figure 10 - USB Pen Drive Technology Model

d. Select DEA Model

The overall function of pen drives is to provide storage to the end-user. Since there is an ever increasing need for more and more storage, and that pen drives lowest prices

hover around ten dollars, pen drive manufacturers and customers are striving for larger and larger capacity. Since this is the case, an output-oriented DEA model will be used to perform the analysis.

e. Collect Data

Since this is a basic example, the information for pen drive technology was gathered from www.pricewatch.com on June 9, 2004, with a search string of “USB pen” and “16MB,” “32MB,” and “128MB” depending on the size of the pen drives being looked for. These results were then filtered further. The prices of pen drives were chosen excluding shipping cost. Shipping and handling costs were excluded because it is not a result of the technology but the distribution mechanism. The subset of data to be analyzed in this example is provided in Table 12.

Table 12 - Sample Dataset of USB Pen Drives

<i>DMU</i>	<i>Manufacturer</i>	<i>Cost (US \$)</i>	<i>Capacity (MB)</i>
A	Targus	12	16
B	Generic	10	32
C	Generic	8	32
D	Generic	25	128

f. Analyze Technological Progress

The data was analyzed with the Gnu Linear Programming Kit (GLPK), php-glpk, ChartDirector 3.10, and the phpTFDEA toolkit, and plotted in Figure 11. In this figure it

is apparent that product D, the generic 128 MB USB pen drive delivers the best cost

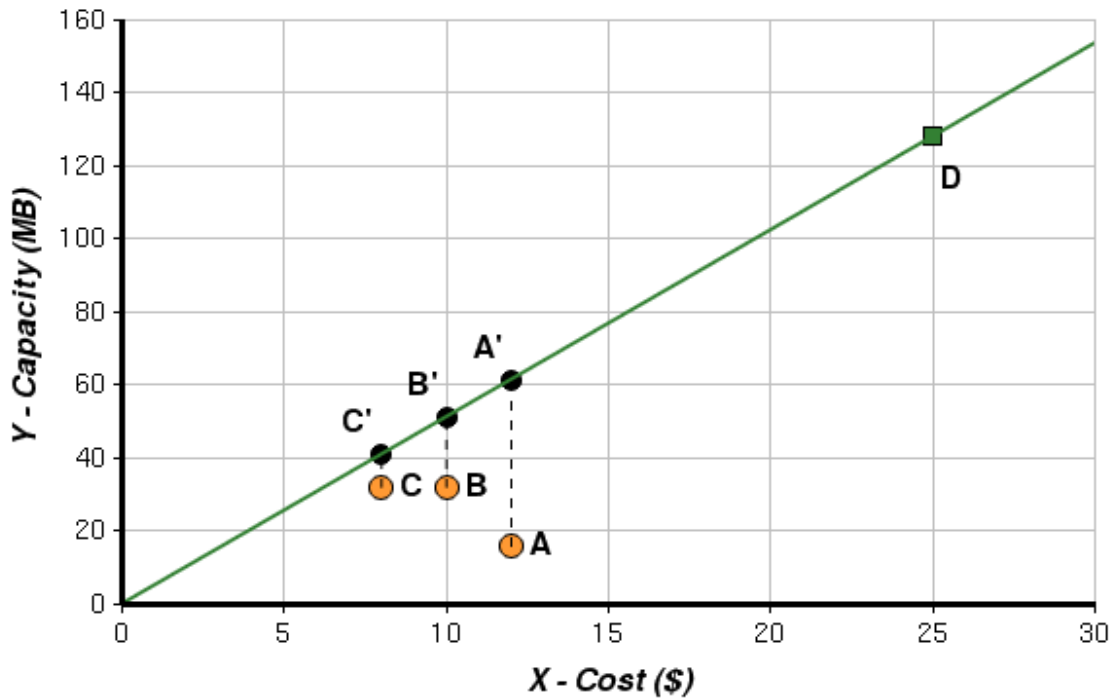


Figure 11 - USB Pen Drive Efficiency Frontier

performance ratio. Those products that do not deliver the same ratio are inefficient. For them to be efficient, it is necessary for them to produce the output donated by X' , where X is the DMU in question.

For A to be efficient the expected output based on the Capacity/Cost achieved by D is calculated in (25).

$$12 \times \left(\frac{128}{25} \right) = 61.44 \quad (25)$$

Using (16) to determine the product A's relative efficiency, the amount of output expected for A to be efficient is determined in (26).

$$\phi_A = \left(\frac{61.4}{16} \right) = 3.84 \quad (26)$$

Repeating this procedure for each product, the DEA CRS efficiencies are presented in Table 13.

Table 13 - USB Pen Drive Output-Oriented CRS DEA Efficiencies

ϕ_A	ϕ_B	ϕ_C	ϕ_D
3.84	1.60	1.28	1.00

g. Examine Results

Upon analysis, two important issues were overlooked in the first model. Given the dataset, it is not possible to purchase a USB pen drive less than C the cheapest of the USB pen drives. This implies there is a cost of entry to purchase any product of this technology which would correspond to an IRS DEA model. Likewise, it is not possible to get more than 128 MB of storage in a drive, reflecting the need for DEA using DRS because the performance/cost for D does not scale to higher levels. Since the systems shows both DRS and IRS tendencies, this application is a candidate for VRS.

h. Improve Model

Using the previous insights, the model is modified and rerun providing the results illustrated in Figure 12. The resulting frontier promotes C to the frontier, while A and B still remain inefficient. The product best representing product A on the frontier is composed of the outputs from product C and D as determined by (27) and (28) and is

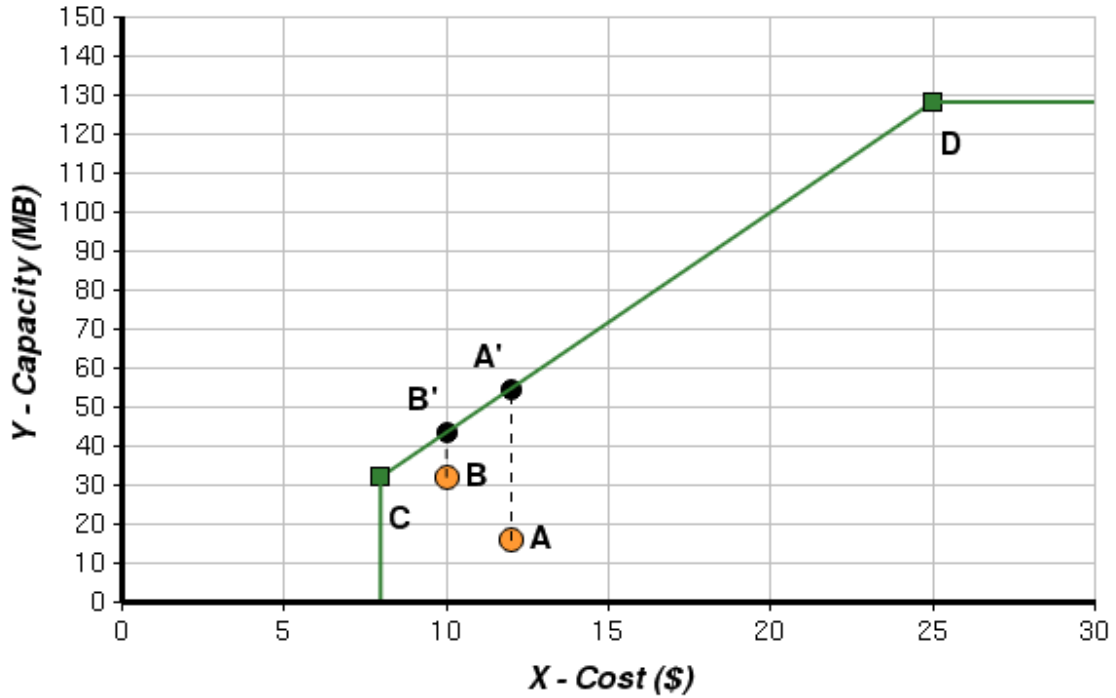


Figure 12 - USB Pen Drive Output-Oriented VRS Frontier

represented by A' in Figure 12.

$$y_{A'} = \left(\frac{25-12}{25-8} \right) y_C + \left(\frac{12-8}{25-8} \right) y_D \quad (27)$$

$$y_{A'} = (0.765)y_C + (0.235)y_D = 54.588 \quad (28)$$

Taking this and combining it with the concept of efficiency in (29) reveals an efficiency of 3.412. In other words, one would expect 3.412 times the output from product A for it to be considered efficient.

$$\phi_A = \left(\frac{54.588}{16} \right) = 3.412 \quad (29)$$

Although this two-dimensional example can be easily examined graphically and

analyzed with basic geometric equations, more sophisticated analysis is required for more complex applications.

7. Critical Review of DEA Literature

Although DEA has been used to assess technology products [11][18][33][34][58], it has not been used to track technological change for products that maintain a constant level of performance. The Malmquist productivity indices mentioned earlier evaluate multiple observations of single DMUs to measure the relative efficiency change [38]. Common applications include banks [24], hospitals [40], and pharmacies [41]. In cases where product performance methods remain constant, these methods fall short [5]. The major research issue in temporal DEA is that there are no methods to assess a product over multiple time periods which is displayed in Figure 13.

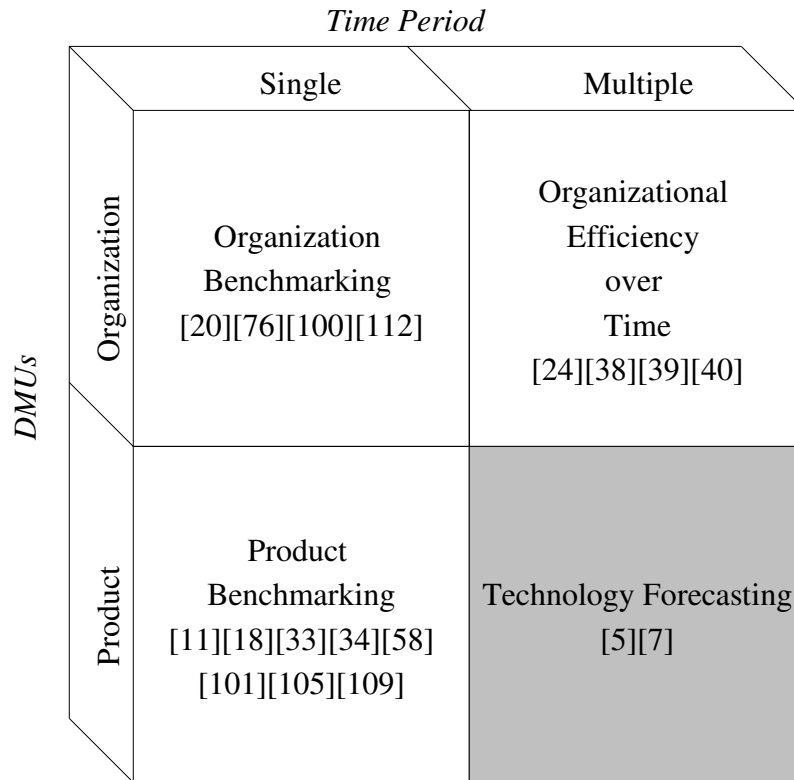


Figure 13 - Applications in Technology Evaluation with DEA

This figure reveals a set of issues that are present in current temporal DEA research which are summarized in Table 14, and discussed subsequently.

Table 14 - Current Issues with Temporal DEA

Issue	Malmquist
a:Allowing DMUs with only one observation	no
b:Allowing irregular time periods	no

a. Allowing for a Single DMU with Only One Observation

Technology products differ from many DEA studies because their performance remains constant barring wear and tear or overhaul. As time passes, there is little change to a product as they are replaced by new ones. In the case of microprocessors, a processor performing at a certain level will always perform at that level.

b. Allowing Irregular Time Periods

One of the main requirements of temporal DEA is that of regular periodicity. DMUs presented in irregular periods are not accounted for. This has broad implications for technology because products are not always introduced at regular periods.

c. Summary

Figure 13, and Table 14 illustrate that current temporal DEA models do not provide for two important characteristics of technology and its products. The first issue is that temporal models do not currently permit DMUs to be observed only once as is the case with new products. Current temporal DEA is appropriate for evaluation of organizations over time, but it falls short when evaluating products which are only introduced once. In addition, current temporal DEA models only permit analysis using regular time periods, which does not necessarily fit the nature of technology.

D. Critical Review of Literature Gaps

Throughout the literature there are a number of tools currently used for technology

forecasting. The purpose of the first section was to provide context to the methods currently available to the technology forecaster through both quantitative and qualitative means. The focus is then narrowed to extrapolative techniques. Extrapolative measures provide a method to predict future trends based on past performance, and provide forecasters an insightful, exploratory tool to determine the capabilities of future products. Challenges currently faced by extrapolative forecasters often revolve around the nature of technologies and the conflict with tools used to forecast it. A number of observations are presented in the preceding sections from which a set of notable methodological gaps using conventional means are illustrated in Table 9 and Figure 13. These research issues include allowing changing trade-offs, and tracking them over time. The primary premise of this research is that DEA can be used to address many of these shortcomings, but there is currently a major gap for the implementation of DEA as a technology forecasting tool.

1. Extrapolative Technological Forecasting with Trade-Off Surfaces

Within extrapolative technological forecasting there are a number of limitations and gaps which are not addressed by current methodologies. These gaps are primarily based on the limitations of current conventional methods in specifying the general form and techniques to represent that form as presented by Gordon and Munson [46]. Figure 14 illustrates the mapping of research issues to gaps in the literature presented previously. The following section explains these issues and maps them to current literature gaps.

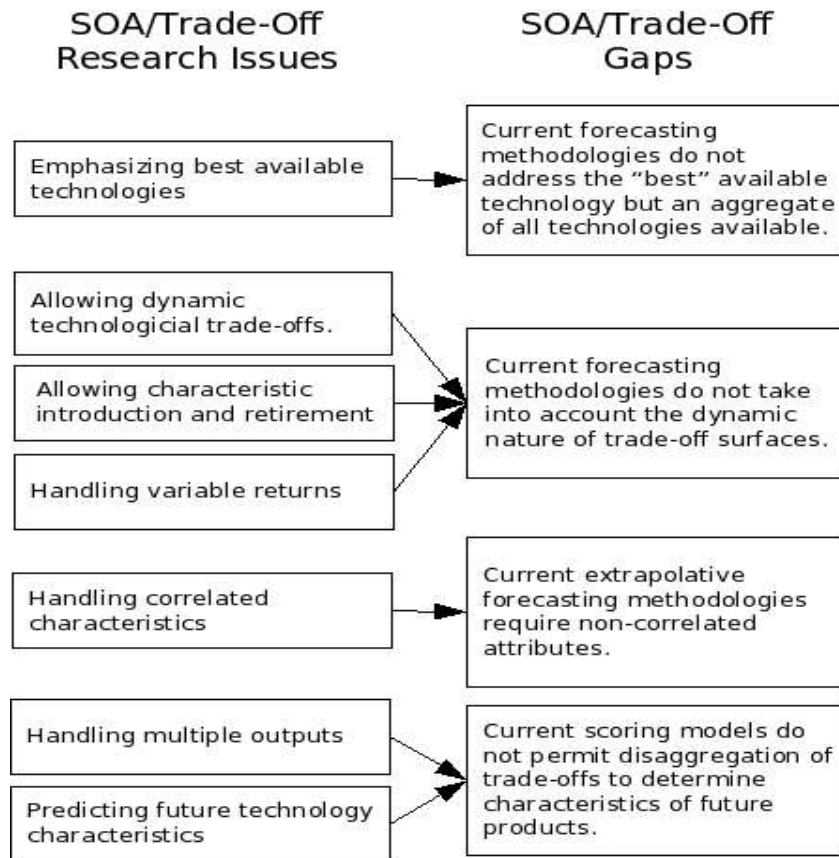


Figure 14 - Technology Forecasting Research Issues Mapped to Gaps

a. Emphasizing the Best Technologies

By its very definition, SOA represents the best available technologies producible at a given time [31]. Current tools for extrapolative technology forecasting use methodologies which rely on trends determined by central point tendencies, which attempt to mitigate the impact of outliers. For many applications, this reasoning is justified. However, technology forecasting is often performed to assess the future limits of technological capability. Contrary to other fields, these limits are nearly always represented by the outliers. Disruptive technologies are often those which are disregarded by manufacturers and designers which rest just beyond the horizon of the

current application norms [26]. By disregarding the outliers, technology forecasters using these methods risk disregarding potential disruptive technologies by design. In his popular text *The Innovator's Dilemma*, Christensen points out that a majority of the disk drive manufacturers have gone out of business due to this disregard for potentially disruptive technologies [26]. This is summarized as gap #1 and illustrated in Figure 14.

Gap #1:	Current forecasting methodologies do not address the “best” available technology but an aggregate of all technologies available.
----------------	--

b. Dynamic Nature of Trade-off Surfaces

By using central-point tendencies, regression and scoring do not allow for time independent fluctuation of trade-offs. Regression analysis, provides an estimate of what the fixed trade-offs will be from past trends to the foreseeable future. Scoring models are similar in that they set a pre-determined value chain to the overall technological index. This forces the assumption that one characteristic is and will forever be more important than others. A popular example of is stressing the gigahertz ratings of processors over power consumption and second level cache. For laptop applications power is arguably more important than gigahertz. By setting predetermined, inflexible trade-offs, the forecaster limits themselves to an very focused set of technological approaches, and does not expect them to change over time.

Another issue which is similar is the introduction and retirement of important

characteristics. Once the important characteristics are set, current methodologies do not allow for new characteristics to be added or removed. This limits the time horizon of many techniques because it is not able to take into account historical data for which it does not have the appropriate data. Thus reducing the applicability of past trends to the future.

Finally, current methodologies fail to take into account a variable returns to scale for technologies. That is, there may be diminishing returns for the advancement of a technology over time. Although three very separate issues, they may be combined in gap #2 summarized below and illustrated in Figure 14.

Gap #2:	Current forecasting methodologies do not take into account the dynamic nature of trade-off surfaces.
----------------	--

c. Handling Correlated Characteristics

Most current methodologies are prone to correlation between measures and requires further analysis or screening to prevent errors. In the event of emerging technologies, it is sometimes difficult for two variables to be inseparable or non-correlated for the purpose of analysis. In the event that a pair of attributes are highly correlated, regression analysis may result in inaccurate results because weighting will be inaccurately biased. A more accurate measure may be formulated by a composite of the values or a selection of only one attribute within the model. However, in the event that such attributes

correlation is altered over time, this further contributes to gap #2 presented earlier, and is summarized in gap #3 and illustrated in Figure 14.

Gap #3:	Current extrapolative forecasting methodologies require non-correlated characteristics.
----------------	---

d. Predicting Future Technologies

Current scoring models do not allow disaggregation of the individual trade-offs to determine future capabilities of technologies. They provide a method to construct an index, but those indices can not be used to assess what the future attributes and characteristics of technologies will be. Although very helpful for assessing the progress of technology over time, this attribute limits the usefulness to many organizations because it is not feasible to determine what the future capabilities of a technology might be, merely the associated score. Regression addresses some of these issues, however it only allows for a single output against which all characteristics are used to determine. Additional issues arise by the necessity for a single rigid output for the technological index. For scoring as well as regression, the output is a single score which is inflexible. This is summarized as gap #4 and illustrated in Figure 14.

Gap #4:	Current extrapolative methods are limited to a single output that can not be disaggregated.
----------------	---

2. Data Envelopment Analysis

The limitations of extrapolative technological forecasting seem to be a prime

candidate for the use of DEA. DEA provides a means to aggregate inputs and outputs, while permitting a number of features that are not obtainable by currently accepted extrapolative technological forecasting methodologies. However, there are currently limitations to DEA which do not permit for a straightforward application of the methodology to technological forecasting. Figure 13 illustrates this in the previous section.

For product evaluation, DEA offers a method to evaluate technologies at a given point in time, but does not extend that to multiple time periods with current temporal models. Instead, temporal DEA models focus primarily on organizations and isolating the success of their management practices from the overall affect of their environment. Because of this, current methods rely on DMUs being present throughout time and changing relative to each other in that time. Technology products however, are introduced at irregular time intervals and are often not improved; meaning that there is little other than the environmental shift which will effect their relative efficiency or technology score. This is summarized as gap #5.

Gap #5:	Current temporal models of DEA do not allow for DMUs which are observed at only one time period and introduced at irregular intervals.
----------------	--

This gap is mapped to the research issues presented in the critical review are displayed in Figure 15.

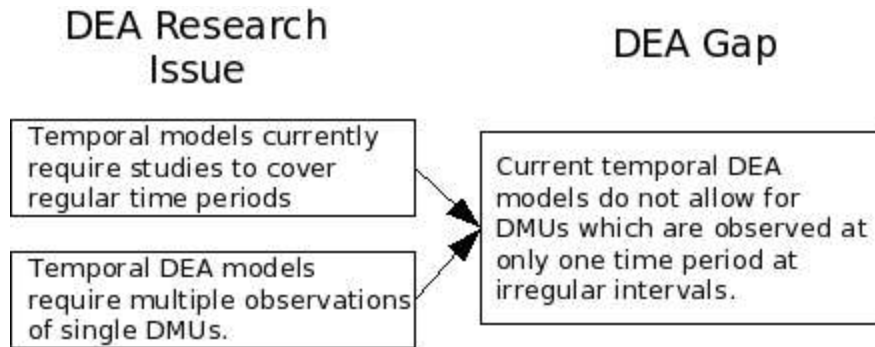


Figure 15 - DEA Research Issues to DEA Gap

3. Literature Gap Summary

The current extrapolative technology forecasting literature has a number of gaps (gap #1-4) that may be resolved through the implementation of DEA. However, DEA itself has a gap preventing its implementation by technology forecasters which once overcome has the potential to provide forecasters with a powerful tool (gap #5). The subsequent sections will address this gap, and combine the two in order to address the previously listed gaps, in addition to creating a new formulation.

III. Research Methodology and Model Development

A. Purpose

The purpose of this section is to present the methodology in constructing and the construction of a new means to forecast technology, technology forecasting with DEA (TFDEA).

B. Methodology Overview

The methodology used is presented in Figure 1 and redisplayed as Figure 16. Stage 1 is presented in chapter II as the literature review. Five gaps are identified related to the issues listed in the critical literature reviews. In this chapter, stage 2 the gaps are used to synthesize a research objective and questions. These questions are then used to create a new model for technology forecasting using data envelopment analysis.

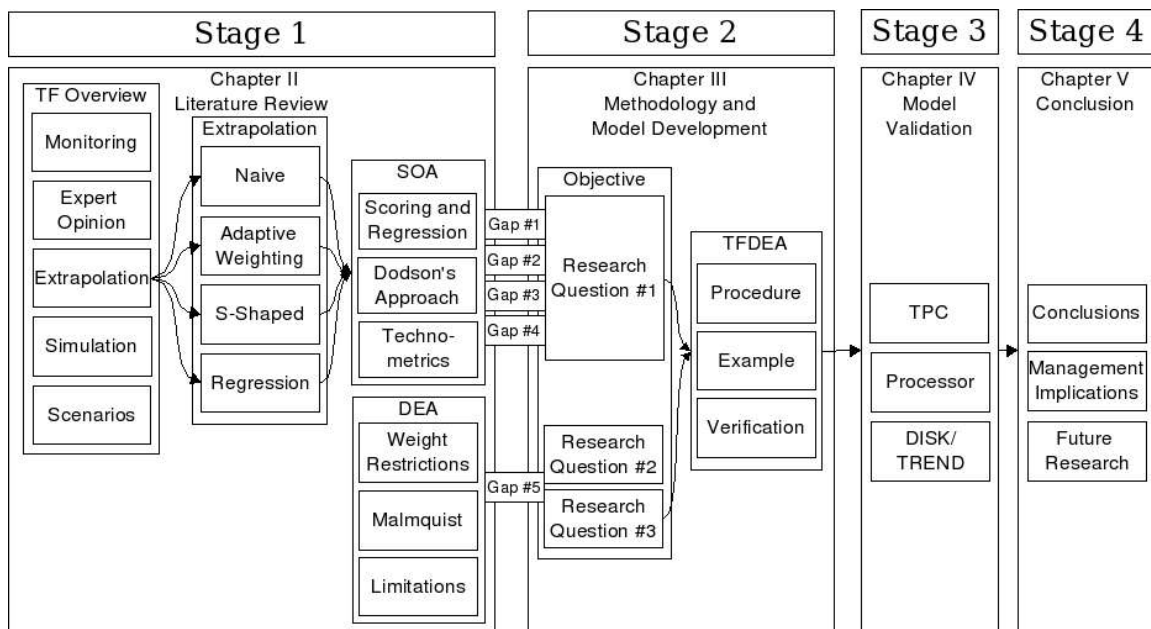


Figure 16 - Research Methodology

In stage 3, Chapter IV, this model is then validated against three case studies. The first study provides a straightforward easily illustrated example using the method. Its primary purpose is to establish basic validity of the model. The purpose of the second application is to extend the model to multiple dimensions to validate its applicability to more complex problems. The final example again extends the methodology to multiple technological approaches and reexamines trends studied by popular management literature. After the validation, the final stage summarizes the overall findings.

C. Research Objective

The application of trade-off surfaces to technology forecasting is a well established practice with a number of gaps as listed earlier. DEA provides a viable methodology to represent these surfaces and overcome many of the gaps we presented in extrapolative techniques to forecast technology.

<p>Research Objective:</p>	<p>Develop a method for technology forecasting which allows for a readily calculable method to measure the SOA and its advance by extending current temporal DEA to allow for DMUs which are introduced once at irregular intervals.</p>
-----------------------------------	--

D. Research Approach

The gaps from extrapolative technology forecasting can be addressed by mapping them to the strengths of DEA as shown in Figure 17. This results in the first research question presented below.

Research Question #1:	<i>How can DEA be used to measure the SOA and trade-off surfaces?</i>
------------------------------	--

While DEA strengths offer potential solutions to currently existing gaps in technology

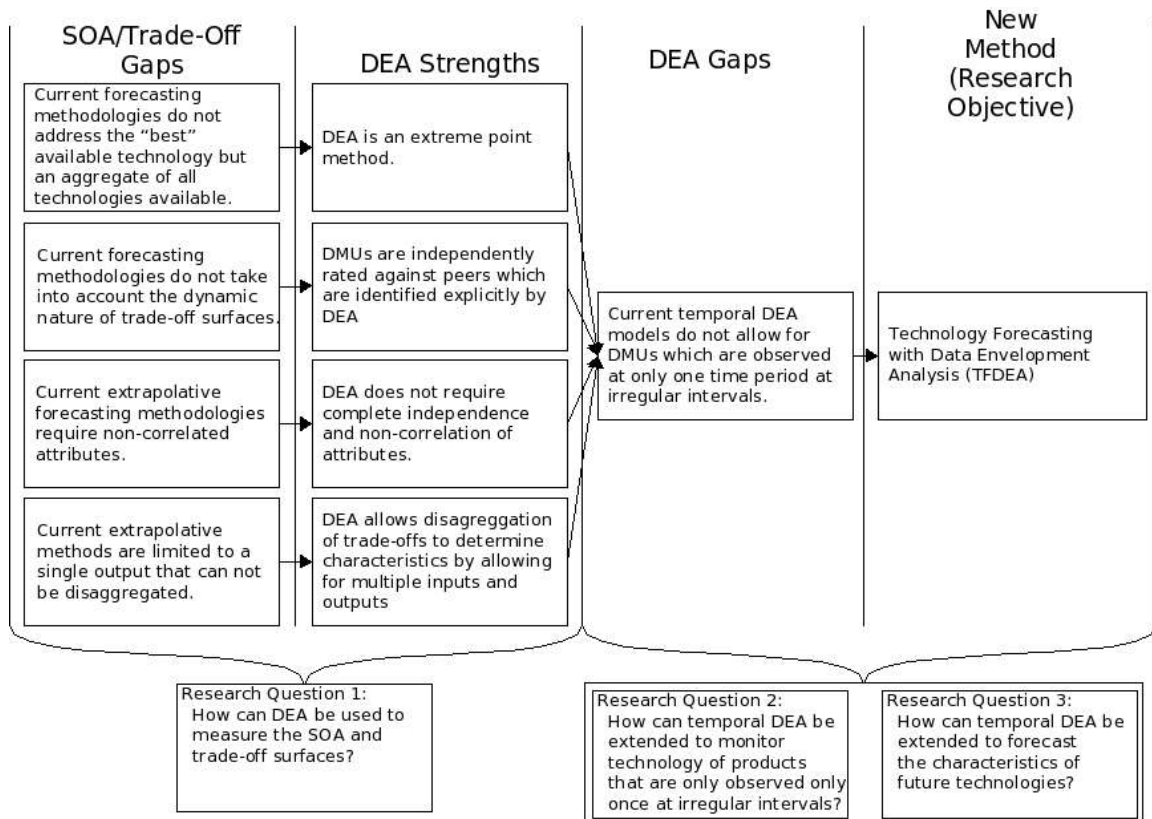


Figure 17 - Research Design

forecasting, temporal DEA does not currently offer a means to monitor technology for products that are introduced at irregular intervals and observed only once. This leads to a second research question, listed below.

Research Question #2:	<i>How can temporal DEA be extended to monitor technology of products that are only observed only once at irregular intervals?</i>
------------------------------	---

This second question resolves the issue of monitoring technology and its progress but

leaves a final question remaining. Once technology can be monitored, there is still a need to predict the characteristics of a useful machine. This is addressed by research question #3, listed below.

<i>Research Question #3:</i>	<i>How can temporal DEA be extended to forecast the characteristics of future technologies?</i>
-------------------------------------	--

E. Using DEA to Represent the SOA

Research question #1 asks how DEA can be used to measure the SOA and technology trade-off surfaces. This can be done by using individual products to construct a DEA efficiency frontier as a representation of the current technological SOA. Using Gordon and Munson's [45] procedure for determining the SOA and combining it with Anderson's [4] DEA methodology, a new procedure will be created in this study as displayed in Figure 18. We have termed the new procedure technology forecasting with data envelopment analysis (TFDEA).

Research question #2 is addressed by the final two steps of the proposed methodology as presented in Figure 18. Research question #3 is addressed by using the information determined by #1 and #2 to determine the future characteristics of products. The following material provides a detailed discussion of the process.

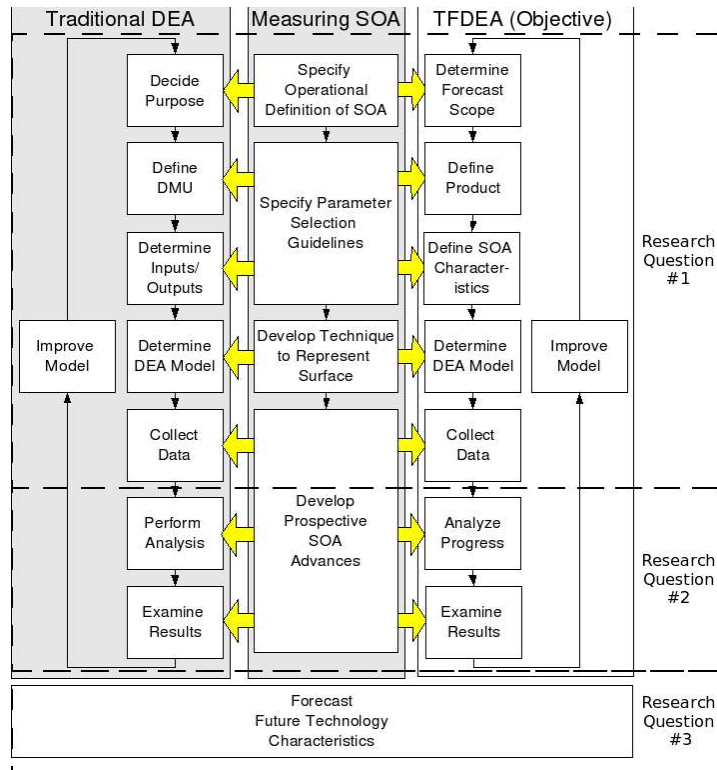


Figure 18 - Map of Combined DEA and SOA Approach

1. Determine Scope of Forecast

A common shortcoming of forecasters is that they look at technology with too narrow of a focus [8]. Conventional techniques often limit scope to particular market segments of a technology [26], which are defined by predetermined preference structures that can limit the forecaster's ability to recognize potentially disruptive technologies. In some cases, the forecaster, or the organization interested in the technology, limit themselves to certain technical approaches. By permitting a broader spectrum of preferences, SOA surfaces allow for a more complete technical analysis.

This research proposes that DEA allows more visibility to an overall technology.

Christensen performed forecasts on single market segments for hard disk drives separately based on reports from DISK/TREND [26][82][87]; however, DEA can look at all segments simultaneously including multiple segments of hard disk drives from laptop computers to desktops to servers. By examining a broader range of technical approaches, those which may be disruptive to the current market trends may be more easily detected, and a better understanding of the technological environment may be achieved.

2. Define a Product

Once the scope of a technology is determined, it is necessary to choose those characteristics to be included in the measurement of a technology. In the example of hard disk drive performance, what elements are to be included in the measurement of that performance? Does this include elements outside of traditional thought such as specific technological elements including disk drive controllers or disk cache? Or, when measuring the performance of microprocessors, are there secondary or off-chip caches to be considered? In other words, is the aim to assess an individual type of technology and its components or the overall function that it provides? Is it an entire system or individual portions of the technology that is to be forecast?

3. Define SOA Characteristics

Using Alexander's classification presented earlier, technology characteristics are either functional or structural [1]. Technology may be considered as the ability of

structural characteristics to deliver functional characteristics. Technological change corresponds to change in the efficiency with which structural characteristics provide the functional characteristics. This readily maps functional characteristics and structural characteristics to the outputs and inputs of DEA models. At any point in time, there are numerous products that define the SOA. An SOA index of technology may be represented by standard DEA efficiencies, θ for input-oriented or ϕ for output-oriented models. By doing this, a forecaster may leverage the ability of DEA to handle multidimensional distance measures not available to conventional means.

The difference between structural and functional characteristics can be illustrated with persistent storage. “Persistent, or non-volatile, storage” is storage used by computers to store data for long periods of time. Common examples of persistent storage include hard disk drives, tape drives, and flash memory. All three provide storage to an end-user which can be measured by capacity, data throughput, and random seek time. Although each type represents different technical approaches to persistent storage technology, common structural elements include volume, power consumption and price. DEA offers a means to collectively analyze these approaches given their universal characteristics independent of user preference.

4. Determine DEA Model

Determination of the specific DEA model is based upon the overall objectives of the

technologies being analyzed. There are a number of decisions to be made when building the SOA model with DEA.

a. Orientation

In cases where the goal of technology is to minimize the structural characteristics required to deliver fixed functional outputs, an input-oriented DEA models should be used. As discussed previously, this coincides with the traditional view of economists [84]. Such a model can also be used to describe the nature of more mature products which are not expected to continually deliver continuously higher functional performance. However, when technologies attempt to maximize functional characteristics for a given set of structural characteristics, output-oriented models are better suited.

b. Returns to Scale

Returns to scale should also be considered when selecting models per the discussion in the section II.C.5.d. The nature of the technology or the segment being analyzed will contribute to the determination of the returns to scale. Additional consideration should be given to the richness of the data set available. Because VRS allows for a larger number of efficient DMUs, a larger set of data is required for the analysis to be meaningful.

c. Virtual Product Projection

In addition, virtual product projection provides a means of making more use out of all observations by projecting non-SOA products to the SOA frontier. When calculating the average rate of technological progress, those regions of the SOA enveloping more products will have a larger influence in the progress calculation. This may result in tighter confidence intervals for the estimated rate of technological change, offering some of the benefit of weighted regression. For a more detailed discussion of weighted regression please refer to [54][71].

Projection of products to the SOA requires a second phase of analysis. Upon release, each non-SOA observation is projected to the SOA using (30) and (31).

$$\hat{x}_{k,i} = \sum_{j=1}^n x_{j,i} \lambda_{k,j} \quad \forall i \in \{1 \dots m\} \quad (30)$$

$$\hat{y}_{k,r} = \sum_{j=1}^n y_{j,r} \lambda_{k,j} \quad \forall r \in \{1 \dots s\} \quad (31)$$

This creates a proxy product located on the SOA surface as of the product's time of release. From this point forward, technological progress is measured against the SOA surface built from both SOA products and proxy products.

5. Collect Data

Collection of data is an ongoing process, which must ensure that all attributes are collected for as many products as possible, including the time each product is released.

6. Analyze Technological Progress

As has been discussed in previous chapters, one major issue facing those forecasters trying to use the concept of the SOA is that the measurement of the frontier is difficult to perform. The disadvantages with current methods include fixed trade-offs and attribute independence as well as *a priori* preference structures. DEA is a well tested methodology that can overcome several of these obstacles by mapping the SOA surface to the DEA efficiency frontier, effectively turning the efficiency score into a “technological index” relative to the position of the SOA.

a. Mapping Technological Progress

The technological index can be represented by the DEA score $\theta_k^{t_k}$ or $\phi_k^{t_k}$ for product k 's position relative to the SOA surface at the time of its release, t_k . Borrowing notational constraints from Dodson, (14), and Malmquist, (21), the output-oriented rate of technological change of product k may be described by (32).

$$\left(\gamma_k^{(t_k+1)} \right) = \frac{(\phi_k^{(t_k+1)})}{(\phi_k^{t_k})} \quad (32)$$

The technological change of product k relative to the SOA at time t_k is represented by $\gamma_k^{(t_k+1)}$. In instances where progress has occurred, $\gamma_k^{(t_k+1)}$ will be greater than one. In the event that $\gamma_k^{(t_k+1)}$ is less than one, there has been technological regress. In the event

that the model chosen is input-oriented, (33) represents the rate of input-oriented technological change as $\beta_k^{(t_k+1)}$. In contrast to $\gamma_k^{(t_k+1)}$, a value for $\beta_k^{(t_k+1)}$ less than one indicates technological progress because less inputs are needed to produce similar outputs.

$$\beta_k^{(t_k+1)} = \frac{(\theta_k^{(t_k+1)})}{(\theta_k^{t_k})} \quad (33)$$

This can be expanded to an exponential form to allow for continuous time intervals by combining (32) and (33) with (3) to become (34) and (35).

$$\phi_k^t = (\gamma_k^t)^{(\Delta t)} \phi_k^{t_k} \quad (34)$$

$$\theta_k^t = (\beta_k^t)^{(\Delta t)} \theta_k^{t_k} \quad (35)$$

For these equations $\Delta t = t - t_k$, where t represents time. Since we are interested in the shifting of the SOA frontier, only those products that were considered SOA at the time of release will be considered, i.e., their $\theta_k^{t_k}$ or $\phi_k^{t_k}$ is equal to one at the time of their release, t_k . Solving for γ_k^t and β_k^t , equations (34) and (35) may be rewritten as (36) and (37).

$$\gamma_k^t = (\phi_k^t)^{\frac{1}{\Delta t}} \quad (36)$$

$$\beta_k^t = (\theta_k^t)^{\frac{1}{\Delta t}} \quad (37)$$

The overall rate of change (ROC) may be calculated with (38) and (39) as a mean of the indices of technological changes, here $n_{efficient}$ represents the number of SOA observations.

$$\gamma^t = \frac{\sum_{j=1}^l (\phi_j^t)^{\frac{1}{\Delta t}}}{n_{efficient}} \quad \forall l \in \phi^{t_i} = 1.0 \quad (38)$$

$$\beta^t = \frac{\sum_{j=1}^l (\theta_j^t)^{\frac{1}{\Delta t}}}{n_{efficient}} \quad \forall l \in \theta^{t_i} = 1.0 \quad (39)$$

This is done for each time period present in the data set following the procedure outlined in Figure 19.

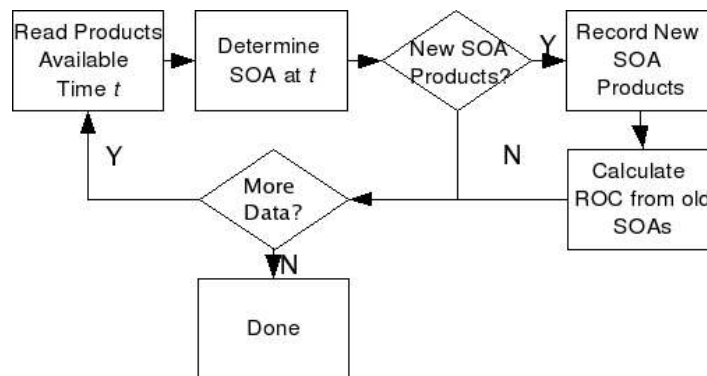


Figure 19 - Procedure for Mapping Technological Change

b. Time Considerations

There are two primary methods to calculate the interval between the current SOA and

the time a product was introduced. The first method is the easiest. The time is considered the same for all observations currently on the frontier. In the event that an SOA surface is defined by products from different time periods, the effective time interval between product introduction and the SOA, t_k' , can be described by (40).

$$t_k' = \frac{\sum_{j=1}^n (t_j - t_k) \lambda_{k,j}}{\sum_{j=1}^n \lambda_{k,j}} \quad (40)$$

Here t_j and t_k represent the times when SOA product j and now non-SOA product k were released. The weight of the reference observation j on the efficiency score, θ or ϕ , of product k is denoted by $\lambda_{k,j}$. In the case when the DEA model is VRS, the sum of all λ is unity reducing (40) to (41).

$$t' = \sum_{j=1}^n (t_j - t_k) \lambda_{k,j} \quad (41)$$

c. Forecasting Future Technologies

Once γ or β has been determined, future SOA frontiers may be calculated by multiplying either SOA outputs (Y_{SOA}) or the SOA inputs (X_{SOA}) by the coefficient of technological progress γ or β raised to an exponent equal to the number of time periods that have passed. This is displayed in (42) for the output-oriented case, and (43) for the input-oriented.

$$x_{i,k}^t = x_{i,k}^{t_k} \times (\beta)^{t-t_k} \quad \forall i \in \{1, \dots, m\} \quad (42)$$

$$y_{r,k}^t = y_{r,k}^{t_k} \times (\gamma)^{t-t_k} \quad \forall r \in \{1, \dots, s\} \quad (43)$$

Although there are a number of means to establish the pattern of future trends [15][17], this research assumes a fixed rate of change over time.

7. Examine Results

One method to test the forecast is to use earlier data to predict later data. This method of testing is further examined in [17]. For this study, a variation of this method is used to study the forecast validity. For each period when a product defines a new SOA point, the resulting ROC's 95% confidence interval is used to predict future frontiers. The sample data consists of all subsequent SOA advances. At each point the SOA is set, the new SOA products are analyzed once against the lower limit for projected products and once against the upper limit for projected products. If the next SOA product is above the lower confidence limit, then the lower limit is considered to have been accurately predicted.

For the upper boundary of the prediction, however, a method is needed to exclude the new DMU from the dataset so as not to redefine the projected frontier. This can be done through the use of DEA super-efficiency [4][111][113]. In this model, the product k DMU being evaluated has λ_k set to zero. This permits for better than perfect efficiencies (thus $\phi < 1$ or $\theta > 1$) for SOA products.

Although this mechanism provides a means to measure those products which may exceed the SOA surface in many cases, it introduces the potential for infeasible answers [95][102]. Since DEA bases efficiency scores on composites of other products, super-efficient products which do not construct the SOA frontier (super-efficient) may not be possible. Although there are a number of methods to address this issue [95][102], a different approach is proposed herein, as illustrated in the following section. By using both the input-oriented and output-oriented models to assess the position of the SOA relative to the product, it is possible to generate a broader forecasted frontier by projecting both inputs and outputs of SOA products into the future. In essence, both β and γ are used to predict x and y .

Further validity should be confirmed with expert opinion or through repeated analysis of the technology to establish a trend. For testing purposes, the upper bound predictions are considered incorrect if a product is declared super-efficient when compared to the projected SOA.

F. Stepping Through the Process

To better understand the process, a brief example is provided, working from the example of USB pen drives presented in II. C. 6.

1. Determine the Scope of the Forecast

Reexamining the previous example of USB pen drives, the scope of this forecast is to

evaluate the USB pen drive technology over time. By studying the trends it is hoped to forecast future USB pen drives and determine if a scheduled release of a pen drive with certain characteristics is feasible.

2. Define a Product

The product of the technology are USB pen drives; small compact highly portable storage devices which plug into the USB port of a computer and are seen as a disk drive.

3. Define SOA Characteristics

The characteristics measured in this case are capacity (MB) and the cost (US \$). This is the best way to assess the form needed to deliver to the end user.

4. Determine DEA Model

Differentiating form from function, the structural model in this example was presented in figure 9. Although the lowest cost of USB pen drives appears to be stabilizing, their capacity continues to grow exponentially, so output orientation is best. Results from the example earlier indicated that a VRS should be used.

5. Collect Data

For this example the data was collected from www.pricewatch.com and fabricated for illustration purposes. The sample dataset is included in Table 15. Although based off of the www.pricewatch.com data, the dates have been arbitrarily assigned for illustrative purposes.

Table 15 - Sample USB Pen Drive Data

<i>DMU</i>	<i>Product</i>	<i>Cost (US \$)</i>	<i>Capacity (MB)</i>	<i>Released</i>
A	Targus	16	16	2001
B	Generic	14	32	2002
C	Generic	8	32	2003
D	Generic	25	128	2004
E	ETT	40	32	2001
F	Generic	30	64	2002
G	Generic	40	256	2004

6. Analyzing Technological Progress

Once the model has been determined and the data has been collected, technology can be mapped, tracked, and then future periods can be predicted.

a. Mapping Technological Progress

Based on the procedure listed before, the frontier for 2001 data is shown in Figure 20. This frontier dictates that all products must match the performance levels matched by A or E to be considered SOA.

The following year, 2002, introduces a new SOA from which technological progress can be calculated. In Figure 21, A' and E' represent the output-oriented virtual products that would be needed for products costing the same amount to remain SOA in 2002.

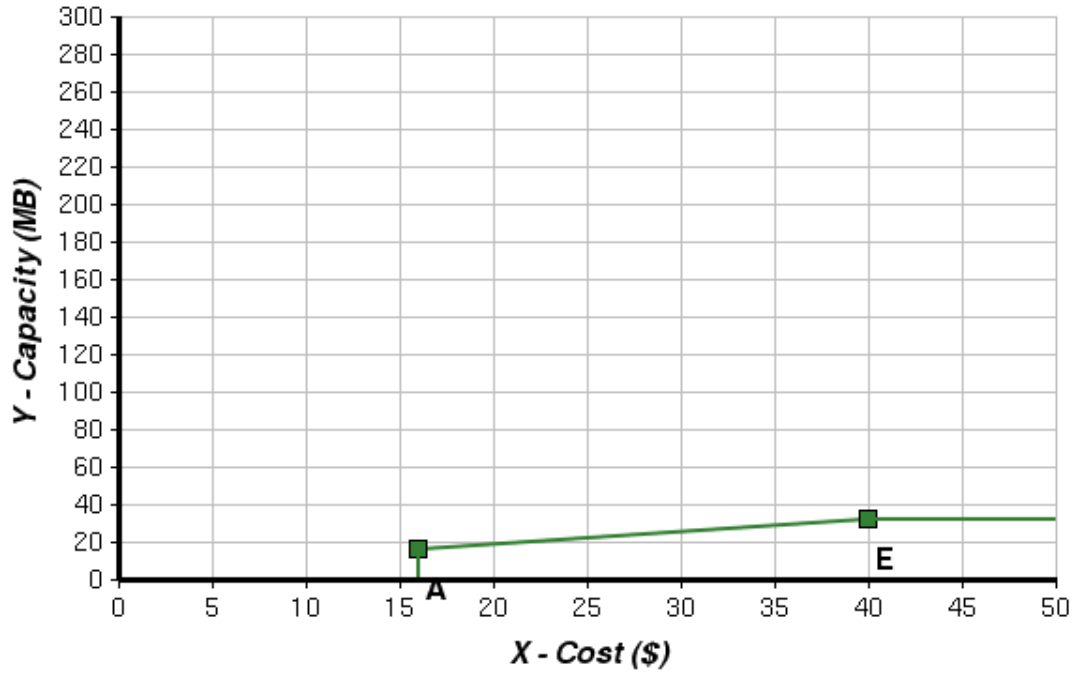


Figure 20 - 2001 USB Pen Drive Output Oriented VRS Frontier

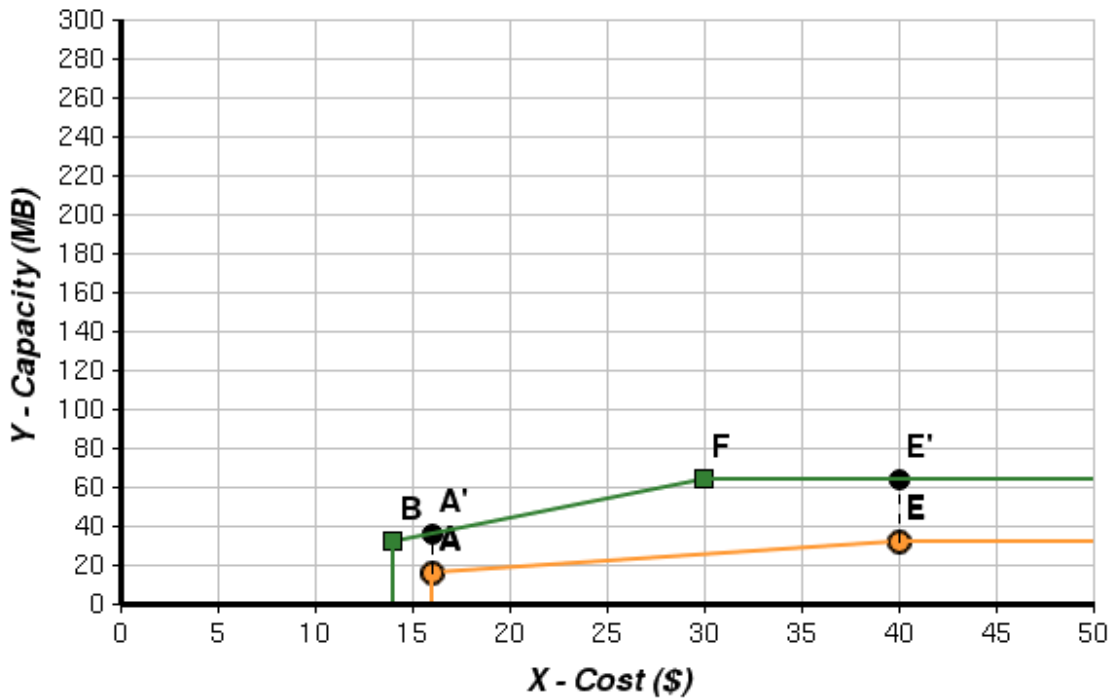


Figure 21 - 2001 and 2002 USB Pen Drive Output-Oriented SOA Frontiers

Using (31), the expected output for product A' is calculated in (44) and (45).

$$y_{A'} = \lambda_{A,B} y_B + \lambda_{A,F} y_F \quad (44)$$

$$y_{A'} = \left(\frac{30-16}{30-14} \right) 32 + \left(\frac{14-12}{30-14} \right) 64 = 36.00 \quad (45)$$

The performance expected for E can be no higher than F resulting in an expected output of F' equal to 64 MB.

These outputs can be used to derive ϕ^{2002} with (46) and (47).

$$\phi_A^{2002} = \frac{36}{16} = 2.25 \quad (46)$$

$$\phi_E^{2002} = \frac{64}{32} = 2.00 \quad (47)$$

The rates of technological change for A and B are calculated using (36), which is shown in (48) and (49).

$$\gamma_A^{2002} = (\phi_A^{2002})^{\frac{1}{(2002-2001)}} = 2.25 \quad (48)$$

$$\gamma_E^{2002} = (\phi_E^{2002})^{\frac{1}{(2002-2001)}} = 2.00 \quad (49)$$

Using (38), the overall rate of change γ^{2002} may be calculated with (50). This indicates the growth of outputs with a given input to remain SOA over time. With a

$$\gamma^{2002} = \frac{2.25+2.00}{2} = 2.13 \quad (50)$$

calculated rate of change, future SOA can be predicted. In this graphical example, the

width of the 95% confidence interval for \bar{y} is assumed to be ± 0.2 . Later cases with larger sample sizes will calculate this through traditional statistical methods. It should be noted that this mean is that of the ROC and not of the actual location of the performance characteristics of a product. This induces a tougher constraint on future forecasts based on this mean.

b. Forecasting Future Technologies

With the rate of change calculated in 2002, a forecaster may use this information with the current SOA frontier to determine the potential future frontiers. Using a confidence interval determined through the use of statistical analysis of the rates of change a range of technology potential is created by an envelope. Starting from the 2003 SOA frontier as defined by C and D, a predicted frontier can be generated using the results of (42), (43), and (50) in conjunction with (34). The boundaries of the potential future SOA are calculated in (51), (52), (53), and (54).

$$y_{B',lower}^{2003} = 32 \times 1.93^{(2003-2002)} = 61.76 \tag{ 51 }$$

$$y_{B',upper}^{2003} = 32 \times 2.33^{(2003-2002)} = 74.56 \tag{ 52 }$$

$$y_{F',lower}^{2003} = 64 \times 1.93^{(2003-2002)} = 123.52 \tag{ 53 }$$

$$y_{F',upper}^{2003} = 64 \times 2.30^{(2003-2002)} = 149.12 \tag{ 54 }$$

This is graphically represented in Figure 22.

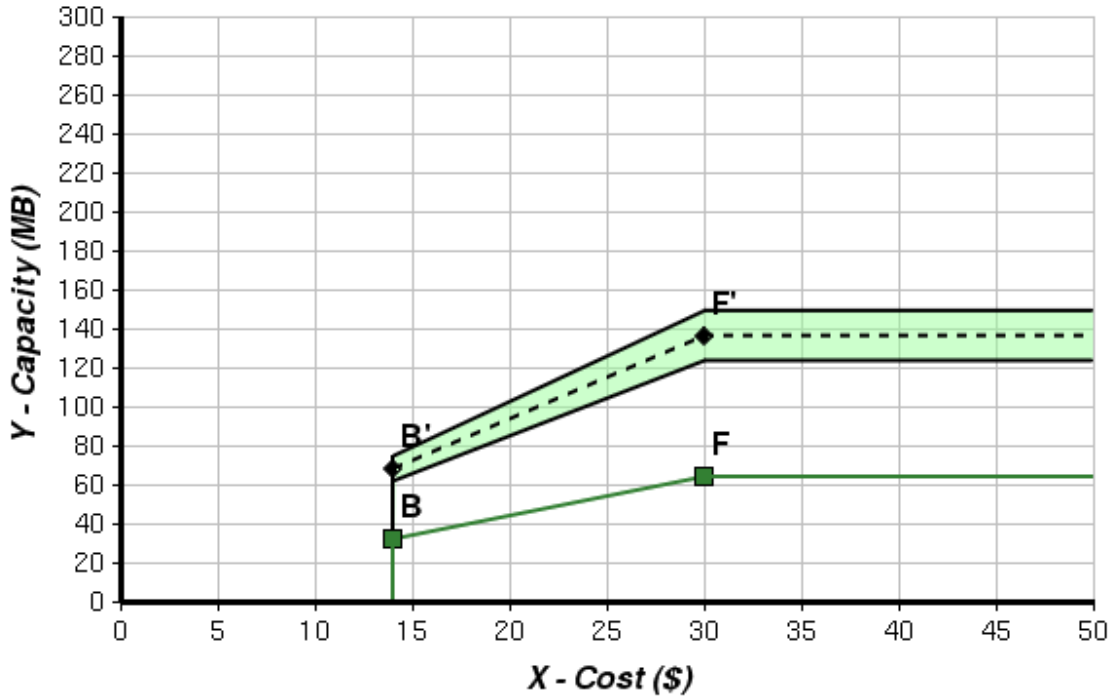
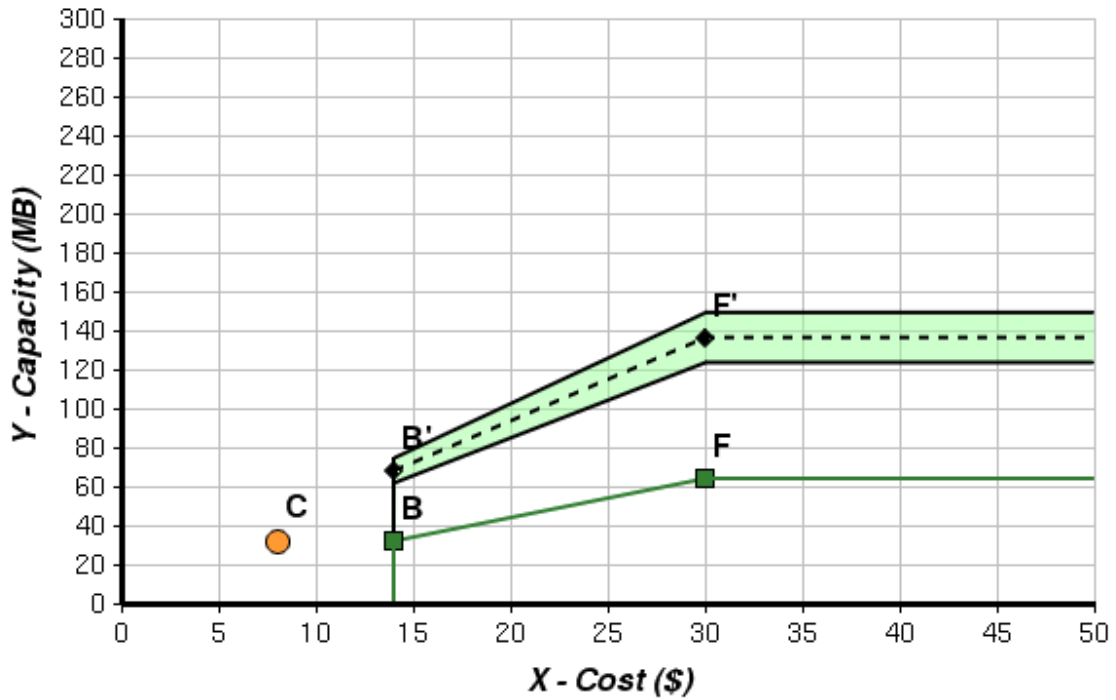


Figure 22 - 2003 Predicted USB Pen Drive SOA Using 2002 Data

7. Examining Results

To test the forecast, data from future points are to be compared against the predicted future. The 2003 data is plotted with the 2002 data and the forecast frontier in Figure 23. Examination of Figure 23 illustrates that product C redefines the technological frontier, but is out of the range predicted by the previous time period. As discussed earlier, using super efficiency to validate forecasts allows the forecaster to assess the frontier by assuming that $\lambda_k = 0$.



When using (31) there is no combination of DMUs present that can construct product C. This instantiates itself as an “infeasible” solution for DEA. To address this, both input-orientation and output-orientation can be used simultaneously. The resulting frontier is illustrated in Figure 24. In this figure, B'' represent the virtual DMU using the input oriented rate of change β with product B. The output oriented model renders the F'' projection as no longer efficient and thus why it is not present in the figure. In this image, C is clearly within the frontier as projected.

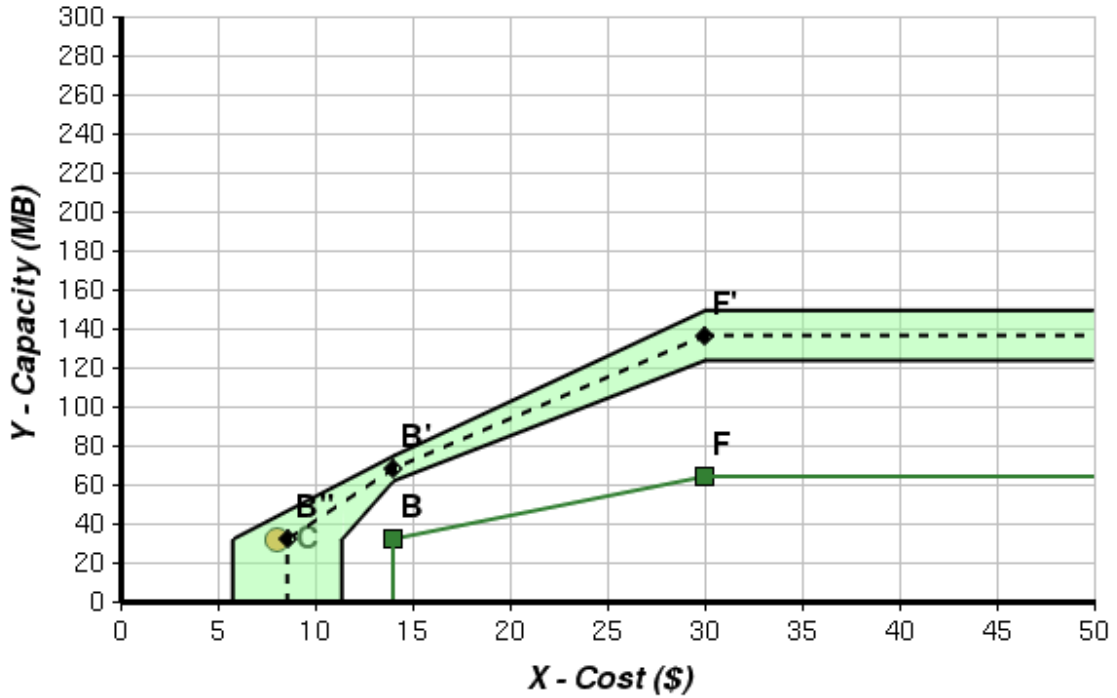


Figure 24 - 2003 USB Pen Drive Frontier Projected with IO and OO

8. Improving the Model

With the actual 2003 data the model may be updated. With the now corrected data the rate of change can be further assessed reflecting the 2003 data. The efficiencies and λ 's of the past references are illustrated in Table 16.

Table 16 - 2003 USB Pen Drive VRS SOA Scores

<i>Product</i>	<i>Release Date</i>	λ_C	λ_F	ϕ
A	2001	0.64	0.36	2.73
B	2002	0.73	0.27	1.27
C	2003	1.00	0.00	1.00
E	2001	0.00	1.00	2.00
F	2002	1.00	0.00	1.00

The illustration of the efficiency frontier in 2003 is illustrated by Figure 25.

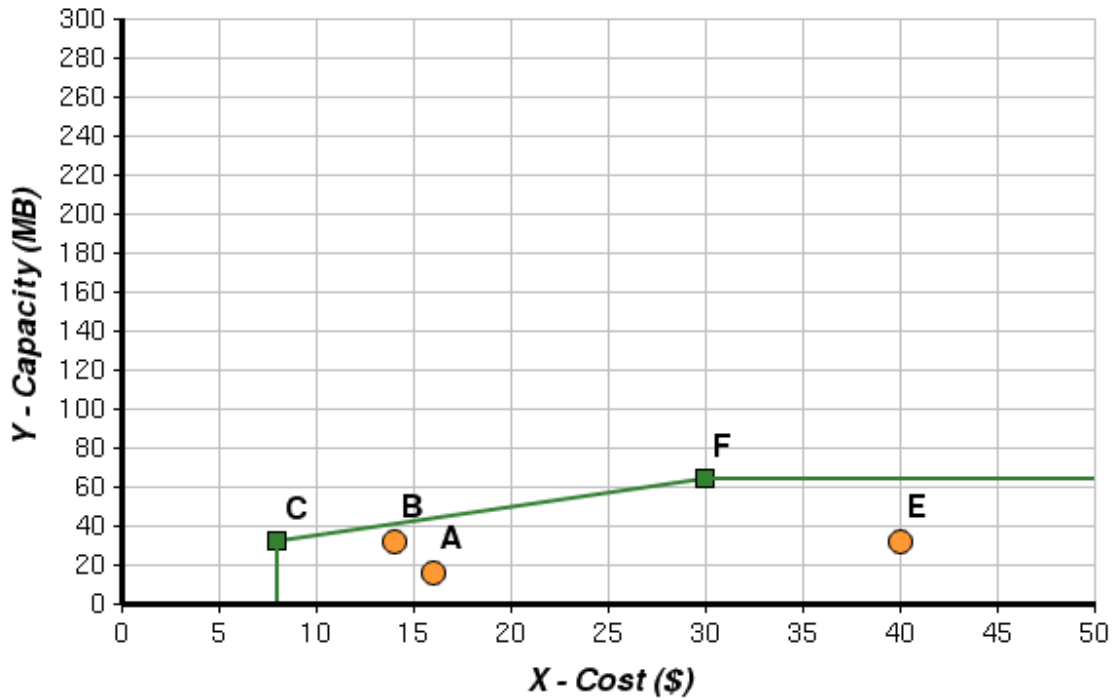


Figure 25 - 2003 USB Pen Drive SOA Frontier with All Products

To calculate the effective time passed between the setting of the new and the old frontiers, (41) is used to calculate the effective time for product A in (55).

$$\Delta t_A' = 0.64 \times (2003 - 2001) + 0.36 \times (2002 - 2001) = 1.64 \quad (55)$$

Repeating this for all previous SOA products, (A,B,E), it is possible to calculate a new rate of technological progress for each product and come up with a mean score with a 95% confidence interval as shown in Table 17. This is plotted in Figure 26 with the products released in the year 2004. In this figure, both products D and G fall within the frontier as predicted by the ROC's 95% confidence interval.

Table 17 - 2003 Output Oriented Rate of USB Pen Drive Technological Progress

Product	$\Delta t'$	γ
A	1.64	1.92
B	0.73	1.39
E	1.00	2.00
μ_γ		1.77
σ_γ		0.33
<i>Confidence interval</i>		0.37

If a product were above the upper limits denoted by the confidence interval, a development team may need to reconsider the date of release or the targeted performance because it is in excess of what one would anticipate is feasible with current trends. On

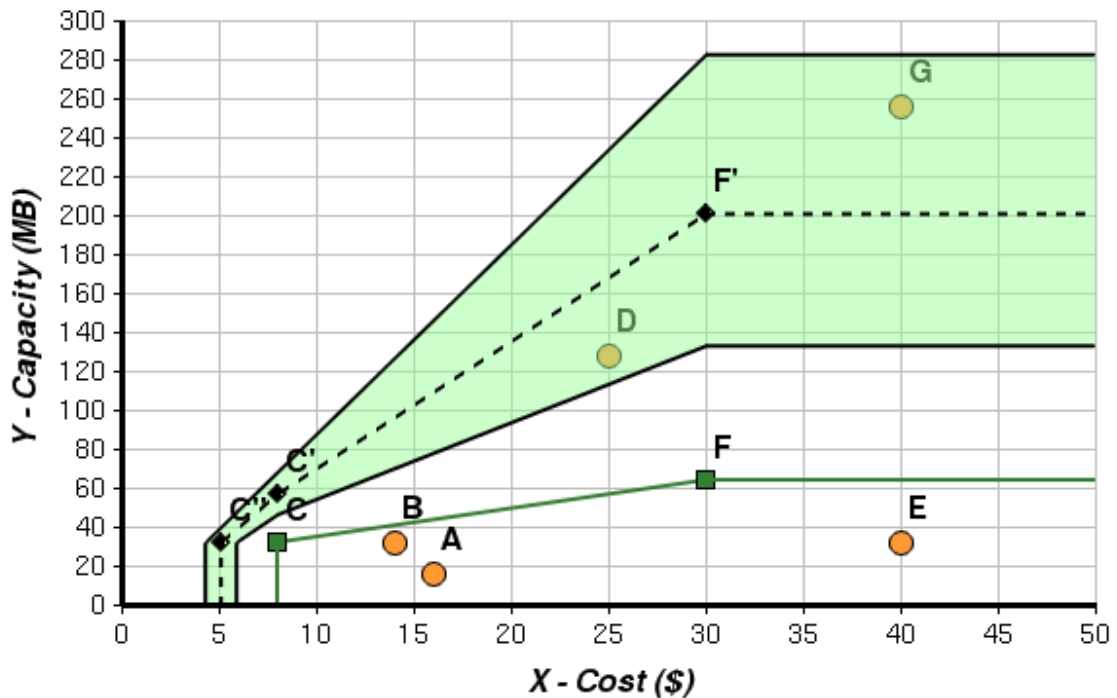


Figure 26 - Predicted 2004 VRS SOA from Previous Data with both Orientations

the other hand if an organization has confirmed performance benefits at this level, there

is reason to believe that it will be a dominant product. In addition, a product group may use this technique in an another way. By determining the super-efficiency of an as of yet un-released product, (40) can be solved for t based on pre-determined γ or β with (34) and (35). For example, if a USB pen drive manufacturer were expecting to release a 512 MB in 2005 costing \$60.00 based on the 2003 SOA; the super efficiency of the desired product would be calculated to be 0.5.

This super efficiency is then used with the expected time of release (2005) with equation and associated (34) which would give an expected release of 2005.7, which is a reasonably close to the estimated release date. With the general models discussed and the process illustrated, verification of the model is required.

G. Model Construction

Due to the large number of calculations required in this process for larger datasets, an application to perform the calculations is required. Although there are many software packages that will perform DEA, the requirements for this application rendered many of them awkward. A secondary goal to keep the tool set affordable to all who would be interested in implementing the method limited the overall budget further. The tool set used for this research included the following software:

1. glpk – Gnu Linear Programming Kit – www.gnu.com
2. php-glpk – glpk extension for php, based on work by blake schweiderman at

www.php4devguide.com.

3. ChartDirector.php – A php chart renderer provided by www.advsoft.com.

H. Model Verification

In order to validate the functionality of the model, verification must be performed.

The point of verification is to confirm that the behavior of the model will go as expected under ideal or known circumstances, therefore being better prepared for real-world applications. For model verification, data is generated with the following methodology.

1. Fixed Rate of Change

The simplest cases used are fixed rates of change, meaning that a constant rate of change is maintained throughout the verification cycle.

- 1) Verification that regular periodic releases at given inputs are accurately predicted.

Given a single data point at $t=0$, and an output rate of change of 1% per time period, observations were created 10000 periods in the future. Those observations were then evaluated at each period to confirm that they were SOA and the rate of change was accurately measured.

- 2) Verification that regular periodic releases at random inputs are accurately predicted.

Once the verification of the model in the simplest case works inputs are varied to confirm that the trade-offs are accurately mapped.

- 3) Verification that irregular periodic releases with fixed inputs are accurately predicted.

Step 1 is confirmed with irregular releases. This ensures that the tool works in less than ideal conditions, which is more closely mimicked by reality.

4) Verification that irregular periodic releases with random inputs are accurately predicted. Step 2 is then expanded to include irregular release dates.

5) Verification that irregular periodic releases with random inputs and multiple observations are accurately predicted. Multiple observations are then added to the results of Step 4.

2. Normally Distributed Rate of Change

The next step is to confirm that the model works for randomly distributed progress over time. One would expect that the predictions hold accurate for varying the inputs and the outputs. We follow the same 5 steps as detailed above.

1) Verification that regular periodic releases and a normalized rate of change with fixed inputs are accurately predicted.

2) Verification that regular periodic releases at a normalized rate of change with random inputs are accurately predicted.

3) Verification that irregular periodic releases at a normalized rate of change with fixed inputs are accurately predicted.

4) Verification that irregular periodic releases at normalized rate of change with random inputs are accurately predicted.

5) Verification that irregular periodic releases at normalized rate of change with random inputs and multiple releases is accurately predicted.

6) Verification that irregular periodic releases at normalized rates of change with random inputs and multiple releases is accurately predicted with less than optimal releases.

Sample Sizes used at current were limited to 300 observations due to constraints of the software used.

IV. TFDEA Validation Case Studies

Three case studies were chosen to validate the new method of technology forecasting. The purpose of these cases is to help refine, illustrate and validate the methodology. The first case study provides a straightforward, easily illustrated two dimensional example used to validate the method with a basic model. The second case study extends the model to six dimensions to assess its feasibility and illustrate its applicability to more complex problems. The final case study provides a more in-depth analysis of a case study in the management technology forecasting literature, and applies the model to multiple technological approaches at once. Each case study follows the steps for performing TFDEA as outlined in chapter III. After the analysis is performed an overview of the results is provided. After all the case studies are presented, a summary of the findings is discussed.

A. Database System Performance: A Two Dimensional Case

The intent of this case study is to provide a straightforward, simple to illustrate example using the newly introduced technology forecasting methodology, TFDEA. By using an easily illustrated model, the methodology can be visually examined as well as tested to confirm the proper behavior.

The first case study evaluates the change in technology of online transaction processing (OLTP) database systems. In the 1980s, automated teller machine (ATM)

networks were one of the first widespread implementations of end-user business transaction automation through information systems. This was expanded to point of sale (POS) devices and registers in retail outlets, gas stations, and grocery stores. Later, Internet powered e-commerce brought transactions closer to the consumer. Unlike traditional batch-processing database systems, OLTP database systems involve live up-to-the-minute transaction processing.

1. Determine the Scope of Forecast

The Transaction Processing Performance Council (TPC) was formed to provide meaningful benchmarks to OLTP users with which to measure OLTP database software. Eventually, the role expanded to benchmark governance to ensure accuracy and integrity [97]. These benchmarks have evolved over time to improve their ability to assess overall database performance and respond to the changing needs of users. As a leading body in the industry and a public source of their benchmarks, TPC sets the standard for measuring the performance of OLTP systems. The scope of this forecast is to assess the technological advances of OLTP systems as defined by their TPC performance.

2. Define a Product

TPC measures performance of products and records the total cost of ownership (TCO) for all software and hardware required to deliver their benchmarks emulating in addition to three years maintenance and operations.

3. Define SOA Characteristics

The current set of performance benchmarks is the TPC revision C (TPC-C) benchmark suite. They measure performance using a combination of five concurrent transactions centered on order-entry activities to provide two measures: overall system cost and transactions per minute – C (tpmC). This example provides a straightforward two-dimensional analysis illustrating the use of DEA to track SOA surfaces over time. Using the structural functional model explained earlier, the specifications required to deliver performance (tpmC) is the TCO.

4. Determining the Specific Model

The functional and structural model of TPC is shown in Figure 27. In this case study OLTP technology represents the ability to transform structure, cost (US \$), into function, performance (tpmC). For the sake of technological progress, effective time will be

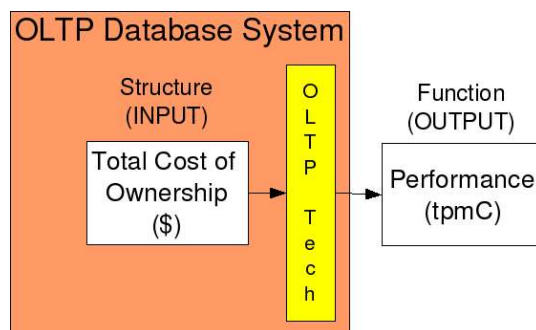


Figure 27 - Basic OLTP Technology Model

calculated.

a. Orientation

Typically, corporate customers of database systems have specific budgets for which they want to maximize their performance. The budget may be a firm or precise value while the future performance needs are more difficult to quantify other than needing to be faster, so an output-oriented model is appropriate in this example. Although there are instances that cost should be minimized, it is not usual that it is minimized based on TPC scores.

b. Returns to Scale

With the TCO including 3 years of operational costs as well as maintenance, there is a requirement for minimum investment in an OLTP system. In addition, once an optimal level of performance is achieved, the cost of incremental performance improvements becomes increasingly expensive. Because of this and the fact that there are maximum levels of performance that are deliverable via the TPC a VRS model has been chosen.

5. Collect Data

The data was collected on March 28, 2004 from www.tpc.org. The data provided by the site includes performance, cost, the date of availability, the date of the test, and other statistics. Manufacturers typically run the tests on products that are not yet available on the market, but are required to record when they are available. The date of availability is the date used to assess the technological progress, because those that are not released are

not available to the general public. The site provides cost in a number of currencies, but only those benchmarks reported using the US dollar were used. The dataset consisted of 130 systems which were released between September 15, 2000 and August 31, 2004.

This data is in the appendix.

6. Analyze Technological Progress

a. General Results

For this study, only those products released prior to the collection date were evaluated to examine the overall rate of change. Forecasts are performed for those which were not available at the date of data collection. The ROC for each OLTP system is calculated for each day the SOA is defined using the effective time passed. These ROCs are then used to determine the frontiers at the time of all future SOAs. These frontiers are then tested against the performance characteristics of the actual future SOA using super-efficiency.

Table 18 summarizes the ROC validation results when used to predict the characteristics of future SOA products. In summary, Table 18 tells a forecaster a number of things. Using the 95% confidence interval of the ROC, \bar{y} , throughout the study period the specifications of SOA products were accurately predicted 42.77% of the time. In other words, those products that defined the SOA fell within the predicted range 40% of the time.

Table 18 - TPC Results of ROC Validation with Output-Oriented Model

Predicted Range	541	42.77%
Above Conservative ROC Estimate	220	17.39%
Below Aggressive ROC Estimate	241	19.05%
ROC Did not Predict SOA	263	20.79%
Total	1265	100.00%

Those products that were not within the predicted SOA range were at least above the conservatively predicted SOA 17%. Of the remaining SOA products, 19% fell short of the aggressive SOA forecasts as determined by the ROC. Products were not predicted when their TCO was lower than anything previously available. This resulted in infeasible solutions with DEA using super-efficiency. There were 1265 attempted forecasts from 130 data points because each time that the SOA was set, all future SOAs were evaluated.

As discussed previously, output-orientation only attempts to manipulate the outputs of products to make them efficient. If there is a drop below the lowest costing product, VRS will not provide a feasible solution. One method to address this is to use both $\bar{\beta}$ and $\bar{\gamma}$ to forecast the future frontier. By doing so, the movement of the efficiency frontier takes into account both the growth of product function and the reduction of product structure. With this increased surface area, the number of infeasible solutions is significantly decreased as displayed in Table 19. Additional effects include the number

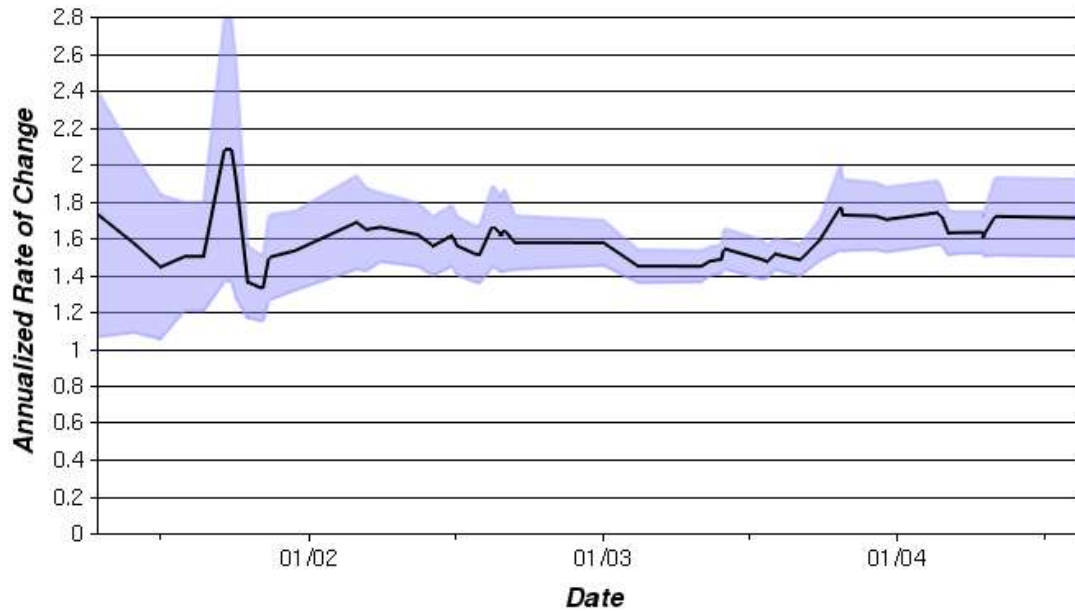
of products which fall within the predicted SOA. The number of predicted SOA product specifications increases to 63%.

Table 19 - TPC Results of ROC Validation with IO-OO Output-Oriented Model

Predicted Range	797	63.00%
Above Conservative ROC Estimate	130	10.27%
Below Aggressive ROC Estimate	338	26.71%
ROC Did not Predict SOA	0	0.00%
Total	1265	100.00%

Those products which are beyond the aggressively predicted frontier compose 10.27% of the predictions. However, the aggressive forecasts of the SOA contain over 89% of the SOA products, providing a guideline for what the limit of future technologies will be. Combined with the range which is accurately predicted, this means that over 89% of the SOA products that are released are within the aggressive forecasts of the SOA.

More importantly, all future products are now predicted using the larger frontier. As a tool for the new product development team, this provides additional insight as to what specifications future products will hold. The annualized mean rates of technological change are presented in Figure 28.



created with ChartDirector from www.advssofteng.com

Figure 28 - Annualized Rates of Change for TPC-C

Table 20 provides the final mean ROCs for March 17, 2004. With these observations, it is possible to test the feasibility of products which have been tested but were not available by the time of the data collection.

Table 20 - TPC Annualized VRS SOA ROC for 03/17/2004

(\bar{y})	1.63
95% Confidence Interval	± 0.12

Table 20 indicates that at a given cost performance is expected to increase at a rate of 63% each year. The confidence interval width on the mean is $\pm 12\%$. The upper confidence interval value of $1.63 + 0.12 = 1.75$ then corresponds to a 75% annual rate of change which will be labeled as the aggressive ROC estimate. Conversely, the lower confidence interval value of $1.63 - 0.12 = 1.51$ then corresponds to a 51% annual rate of

change which will be labeled as the conservative ROC. This methodology is used to predict the future position of the frontier based on the March 17, 2004 data in Table 21. The table shows that the HP Integrity Superdome is on target to match the SOA as per the past trends, meaning that it will be on or near the SOA on release. On the other hand, HP appears to be overly optimistic with its Integrity rx5670 because the super-efficiency, $\phi_{aggressive}$, of the product on the predicted SOA is well below 1. The $\phi_{conservative}=0.82$ for HP's rx5670 indicates that based on a conservative estimate of the ROC in the enterprise database system industry, not only will the HP rx5670 be SOA but it will exceed the expected competition by that time by 18%. This high technical risk for the rx5670 can be explained by it being the first large scaled Linux based database system in the dataset.

Table 21 - Summary of Upcoming Database Releases Based on 2004-03-17 Results

<i>Product</i>	<i>Date</i>		<i>Technical</i>	
	<i>Available</i>	$\phi_{conservative}$	$\phi_{aggressive}$	<i>Risk</i>
IBM eServer xSeries 365 4P c/s	2004-03-31	1.23	1.30	Low
HP Integrity Superdome	2004-04-14	0.95	1.02	Target
HP rx8620	2004-04-15	1.04	1.07	Low
Unisys ES7000 Aries 420 Enterprise Server	2004-04-20	1.20	1.24	Low
HP Integrity rx5670 Cluster 64P	2004-04-30	0.82	0.89	High
PRIMEPOWER 2500	2004-04-30	1.64	1.77	Low
IBM eServer pSeries 690 Model 7040-681	2004-08-16	1.08	1.21	Low
IBM eServer Xseries 445 8P c/s	2004-08-31	1.44	1.59	Low

The HP Integrity Superdome's values for $\phi_{aggressive}$ and $\phi_{conservative}$ enclose 1.0, which indicates that the price and performance are consistent with what is expected to be state of the art when it is released on April 14th, 2004. The rest of the systems will likely be under performers and are at risk of a competing OLTP systems providing more performance at the same cost.

Additional insight can be gained by more closely analyzing the $\phi_{aggressive}$ and $\phi_{conservative}$ results. By doing this, one could conclude that the HP rx8620 is not far from the SOAs lower bounds, so it could very well be on target. The PRIMEPOWER 2500, however, is likely very far from efficient in the future and may be a non-competitive under performer.

For a more general set of predictions, the observations which are SOA at the last date, March 17, 2004, can be extrapolated into the future using both (\bar{y}) and $(\bar{\beta})$ to generate the points as shown in Table 22 and drawn in Figure 29. As discussed in the methodology section, those DMUs projected with output-orientated DEA are denoted by a single ', and those which are based on input-orientated DEA are indicated with a ". In one year the performance would be expected to have increased considerably while reducing the costs, indicated in Table 22. The columns labeled "Conservative ROC" and "Aggressive ROC" are based on the lower and upper bounds of the ROC's 95% confidence interval.

Table 22 - 2005-03-17 TPC VRS SOA Predictions Based on 2004-03-17 Data

<i>Product</i>	<i>Original</i>		<i>Conservative ROC</i>		<i>Aggressive ROC</i>	
	<i>Cost</i>	<i>tpmC</i>	<i>Cost</i>	<i>tpmC</i>	<i>Cost</i>	<i>tpmC</i>
106"	33,692	17,192	18,260	17,192	14,233	17,192
114"	40,775	22,052	24,051	22,052		
121"	65,528	35,030	40,133	35,030	33,876	35,030
121'			65,528	53,099		
95"	223,664	82,226	113,045	82,226	84,838	82,226
95'			223,664	146,701	223,664	180,577
112'	1,384,981	309,037	1,384,981	494,219		
99"	5,105,486	786,646	2,570,065	786,646	1,943,886	786,646
99'			5,105,486	1,408,273	5,105,486	1,722,256

Figure 29 illustrates the forecast in March 17, 2005 and displays that the frontier does not consist of all possible virtual products. In this figure product 99's input and output-

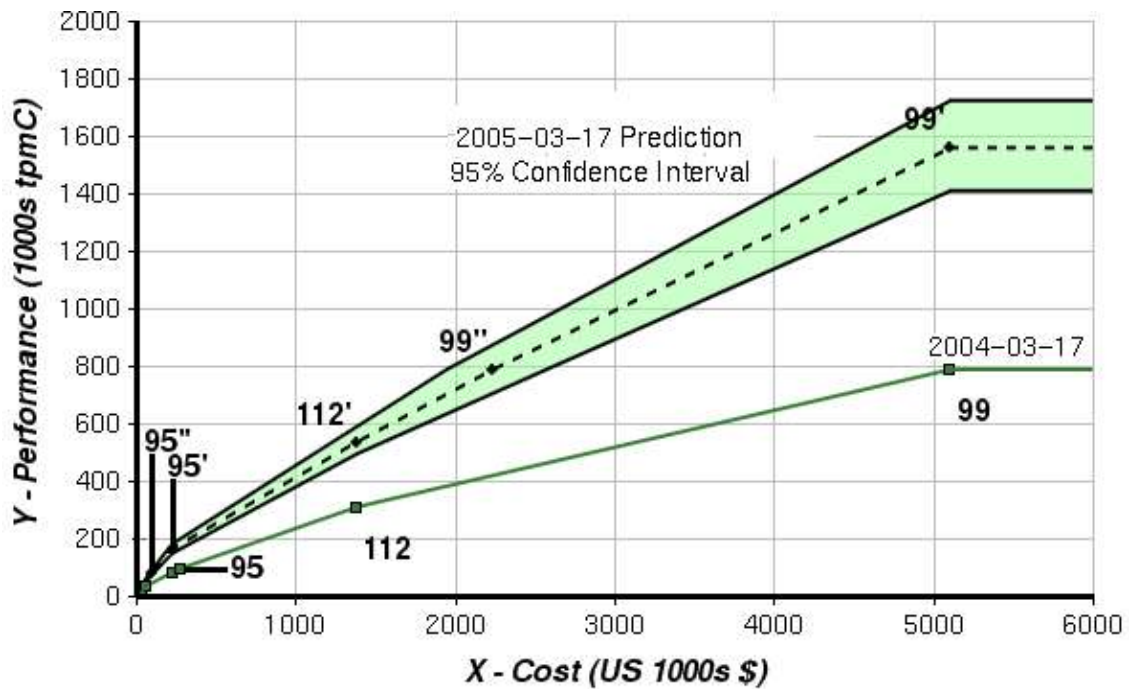


Figure 29 - 2005-03-17 TPC SOA Forecast Using 2004-03-17 Data

oriented virtual products compose parts of the frontier. However, not all products are projected to the frontier because some virtual products are not on the new frontier. In Table 22 this is apparent by the missing values 112', 114', and 106'. Product 112, for example, is only present in the output projection for the predicted frontier because 95' is more efficient.

Using this chart, one would expect the SOA SOA products available on March 17, 2005 to fall within the shaded region. Meanwhile it is certainly likely that there will still be many non-SOA systems available at this date which would lie below the shaded region. Those that aggressive forecasts would be at risk of not making their announcement date or reaching the expected performance.

b. Comparison Regression Results

The RAND technique regresses characteristics as the independent variables against the date as a dependent variable. Applying RAND to this case-study, the performance and cost are mapped to the technology characteristics, and the availability date is used as the dependent variable. To compare the RAND approach to TFDEA; the first 64 data points are used to create the regression equation and the TFDEA SOA frontier on February 12, 2003. The regression equation and the TFDEA ROC is then used to predict the date of availability for the final 66 products assuming that they are state of the art and continue trends established by regression and TFDEA.

The regression analysis of the first 64 data points constructs the equation presented in (56).

$$t^0 = 12/13/2001 + 0.00222 \times tpmC - 0.0000874 \times Cost \quad (56)$$

In this equation t^0 , represents the time of availability for a product with performance $tpmC$ and cost $Cost$. Performing the regression analysis on the first 64 data points constructs yields the results provided in Table 23.

Table 23 - TPC Regression Results 09/15/2000 – 02/12/2003

<i>Attribute</i>	<i>Result</i>
Multiple R	0.5258
R Square	0.2765
Adjusted R Square	0.2528
Standard Error	220.73

Applying this model to the remaining 66 observations to determine their expected date of release and comparing it to their actual date of release results in an average residual of 486 days with a standard deviation of 196 days.

For TFDEA, the frontier and ROC is determined using the first 64 data points. Using this frontier with each of the last 66 data points, a DEA super-efficiency is determined. The resulting score is then used in conjunction with the calculated ROC and (34) to determine the date of expected introduction. This expected date of introduction is then compared to the actual date of introduction.

TFDEA results in an average residual of 135 days with a standard deviation of 115 days, which illustrates a large improvement in the representation of the state of the art. One product, 106, was dropped from TFDEA because it was not within the scope predicted.

7. Analysis of Results

The purpose of this case study is to validate the usage of the model with a straightforward easy to illustrate example. The results themselves showed positive signs that this method is effective. The major lessons learned by this study are detailed below.

a. Double Orientation

The initial run using only output orientation to determine technological progress and test the validity of the forecast generated a number of infeasible solutions. These infeasible solutions were explained by ever cheaper OLTP systems being released. Because the DEA model was output oriented, it could not solve for decreasing minimal costs of entry, and therefore many new products fell out of the scope of the validation forecasts. Using both input and output-oriented DEA (double orientation) to create a frontier resolved this issue. This reveals that there are two potential influences in the OLTP market, one is to reduce the cost and an additional is to increase the performance.

b. Identification of Disruptive Technologies

The method also properly identified a new disruptive technology through the detection

of the HP machine which used Linux as its operating system.

c. Effectiveness of the Forecast

When the double orientation was chosen, every SOA product fell within the validation forecasts. Of the 1265 forecasts for future state of the art, over 60% of them fell within the between the aggressive and conservative estimates of the forecasts. This means that a forecaster has potential to determine the future state of technologies with the method.

d. Comparison to Other Methods

When compared to standard regression, a number of differences are apparent. First, the regression does not provide as accurate a prediction of the date which a product will be released. Next, the fixed nature of cost/performance does not take into account the variable returns to scale that OLTP systems actually face. TFDEA did not suffer these problems and was able to provide a closer fitting surface than regression. Finally, TFDEA is able to provide a future plausible frontier for SOA products. This provides the forecaster a method to ascertain the state of technology in the future based on past performance.

B. Microprocessors: A Multiple Dimensional Example

The previous section presents a straightforward two-dimensional example which can be enriched by the expansion to six dimensions. The purpose of this case study is to determine the validity of the model in multiple dimensions. Moore's law as presented in

Naïve Models on page 36 has become one of the most accurate and influential technological forecasts of the latter 20th century. It has been applied to a wide range of technological innovations and measurements despite the fact that it originally corresponded only to the number of transistors on an integrated circuit, not the microprocessor that had yet to be invented. Over the past 37 years, Moore's law has held with surprising accuracy albeit with gradual lengthening of the doubling period to 18-24 months.

However, Moore's Law focuses on only a single characteristic of technology that is of structural orientation. Users are more interested in performance and not the number of transistors. In order to better map microprocessor technology over time, it is essential to take this and other structural characteristics into account. The number of transistors within ICs is not the only measure of structural progress, and as such other vital elements are needed to better map technology over time. By using a functional and structural model in conjunction with the SOA, a better means of technology tracking is achieved.

1. Determine Scope of Forecast

The original scope of Moore's law included all cheaply produced ICs, prior to the processor. In this study, it shall apply to microprocessor technology as a whole.

2. Define a Product

For the intent of this study, products are considered commodity microprocessors.

3. Define SOA Characteristics

To better map the measurement of SOA in the microprocessor arena, a new functional and structural model mapping microprocessor technology is displayed in Figure 30. This

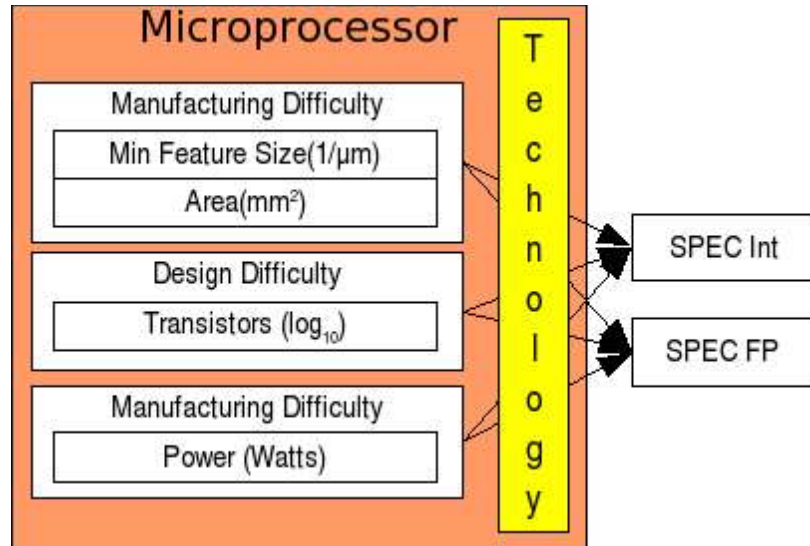


Figure 30 - Enhanced Microprocessor Technology Model

model was originally presented by Anderson [5] *et al.* and later expanded [7]. Looking first at the structural perspective of the model, the inputs reflect the difficulties associated with the manufacturing, design and usage of microprocessors. *Manufacturing difficulty* is represented by the minimum feature size and the total die size. Despite changes in technology, the cost of silicon surface area has remained relatively constant over time, thus resulting in the need for doing more with the same amount of space [56][72]. Since 1971, manufacturing processes have increased in difficulty due to the reduction of minimum feature size from 10μm to 0.09 μm and the increased potential for defects caused by errant dust and other particles. To reflect the fact that smaller feature sizes

correspond to increased difficulty, the feature size must be inverted.

Design difficulty or complexity represents the attribute that Gordon Moore originally addressed with his famous forecast. At the time, adding transistors increased performance. Today, performance is often enhanced through larger pipeline capacities, various cache levels, and the number of registers. All of these are typically done by the addition of transistors. Increasing these attributes complicates aspects of testing and design [53]. The number of transistors directly and indirectly influences other elements that affect design difficulties, including the effects of multiple layers and wire resistance. Since the nature of the growth is exponential and DEA assumes linearity, the number of transistors is \log_{10} transformed.

Usage difficulty refers to the flexibility of a technology and its ability to be adapted to a large number of uses. For microprocessors, power consumption provides a good representation of this flexibility. Although not a major concern for many desktop or server markets, the increase in power and clock speed adds to heat generation. This heat generation affects overall performance and is a major issue for usage in blades and rack-based servers. In addition, low power processors are in increased demand to address the concerns of battery consumption in portable devices.

Functional characteristics or outputs are represented by the speed and ability of microprocessors to run programs that are measured with processor benchmarks. As with

all benchmarking, finding the appropriate means to measure performance is difficult [50]. Early microprocessor performance was measured in millions of instructions per second (MIPS). This method is flawed as a measurement of processor performance, since among many reasons, each instruction does not contribute equally towards performing useful every day tasks. In it's most extreme case, many microprocessors have an instruction referred to as a no-operation that by definition does nothing.

To better measure microprocessors, the Standard Performance Evaluation Corporation (SPEC) was created in 1988 and devoted to “establishing, maintaining and endorsing a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers [99].” Their first benchmarks were introduced in 1989 followed by SPEC92 and then SPEC95. In 1995, SPEC split the benchmark into two benchmarking suites designed to represent two distinct functional units of microprocessors.

Each suite consists of a number of test applications whose results are aggregated and normalized against a reference platform to generate the benchmark. The SPEC95 benchmarks were replaced in 2000 and are about to be replaced again in 2005. For the purpose of this study, the results of the SPEC95 benchmarks will be examined.

4. Determine DEA Model

a. Orientation

Moore's law involves increasing the amount of transistors on silicon, but it is commonly interpreted as getting more processing performance from commodity hardware. Because of this an output orientation has been chosen.

b. Returns to Scale

With a large number of characteristics being measured versus the number of products released, a VRS model would produce a large percentage of SOA products. If more DMUs are efficient, then the ability to measure or even observe changes is reduced. For these reasons, a CRS model has been chosen.

5. Collect Data

Unlike the TPC-C example, in the CPU model there are numerous dimensions of the technology that must be considered. The data set is comprised of 56 processors with their year of release between 1992 and 2000. Over the eight time periods, there are two years that have no new products, and one year with eleven new products (1997). As previously mentioned, a general rule of thumb for performing DEA is to have at least twice as many DMU's as the combined number of inputs and outputs. VRS models tend to have a larger ratio of efficient DMUs to inefficient for the same model, thus requiring a larger data set to draw the same level of discrimination between efficient and inefficient

observations.

6. Analyze Technological Progress

a. General Results

Running the model using a period of a year, the results are shown in Table 24. Unlike the previous case study, TPC, the forecasts are less clear.

Table 24 - Microprocessor Results of Validation with Output-Oriented Model

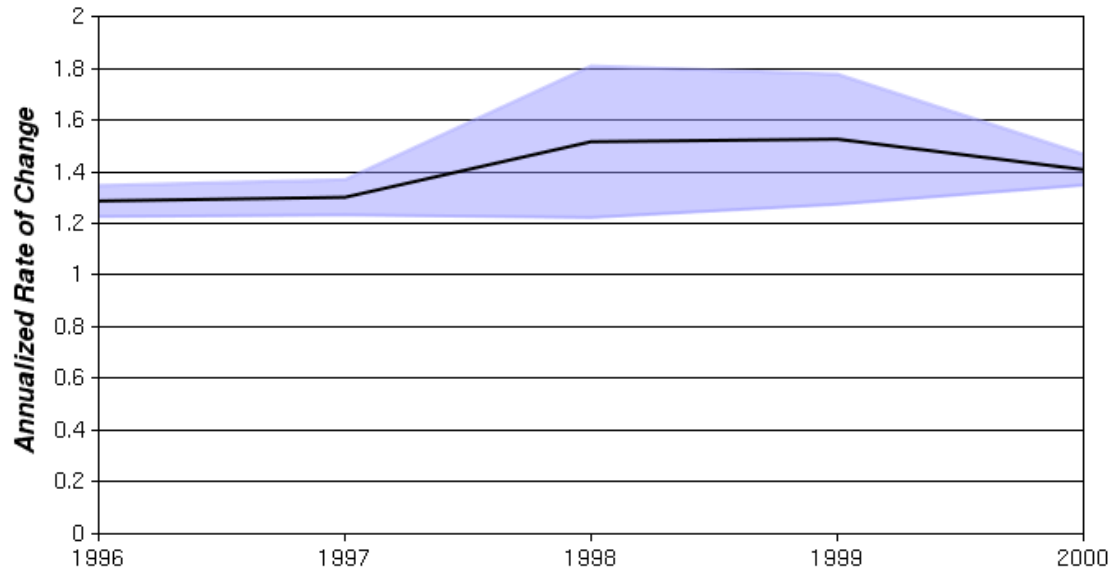
Predicted Range	3	15.00%
Above Conservative ROC Estimate	17	85.00%
Below Aggressive ROC Estimate	0	0.00%
Did not Predict	0	0.00%
Total	20	100.00%

The confidence interval of the ROC only accurately predicts the SOA three times, as indicated by the “Predicted Range” row in the table. The multidimensionality of the problem allows for a significantly larger number of efficient DMUs increasing the ratio of products which can be SOA. Because of this, the number of first time SOA products increases resulting in a smaller number of meaningful SOA frontier shifts to analyze.

With the shorter time window, the forecast has less time to stabilize. However, what can be learned from this model is that the lower bounds of the rate of the change will likely predict the minimum increases in the technology.

A graphic representation of the ROC is presented in Figure 31, which illustrates a

close to constant rate of technological change. Added examination of the results also reveals that the year 1999 missed its one prediction by a fairly small amount. That is, the SOA in 2000 had a super-efficiency score very close to 1, indicating that it is not far



created with ChartDirector from www.adusofteng.com

Figure 31 - Microprocessor Rates Of Change for Output-Oriented Model

from the frontier.

b. Comparison to Moore's Law

Examination of the rates of the change can be further studied to assess the validity of the model for mapping the effective efficiency of observations in relation to Moore's Law. Assuming that products released must be close to the SOA in order for them to be commercially viable, the expected efficiencies of the DEA model will likely be higher than that of the single attribute as described by Moore's Law. Building on the concept of SOA presented by Dodson, it is possible to represent the technological index of a new

microprocessor in terms of the expected number of transistors as presented in (57).

$$\phi_{j, Moore's}^0 = \frac{\text{Moore's Law Expected Number of Transistors}}{\text{Actual Number of Transistors}} \quad (57)$$

In this equation, $\phi_{j, Moore's}^0$ represents the Moore's Law technological index at the time of microprocessor j 's introduction. If TFDEA provides a more comprehensive metric for microprocessor technology, $\phi_{j, TFDEA}^0$ would be closer to unity than $\phi_{j, Moore's}^0$. This can be tested with a basic t-test summarized by the hypothesis listed in (58).

$$\begin{aligned} H_0: & \overline{(\phi_{j, Moore's}^0 - 1)} - \overline{(\phi_{j, TFDEA}^0 - 1)} = 0 \\ H_1: & \overline{(\phi_{j, Moore's}^0 - 1)} - \overline{(\phi_{j, TFDEA}^0 - 1)} > 0 \end{aligned} \quad (58)$$

The null hypothesis, H_0 , is true if the technology indices $\theta_{j, TFDEA}^0$ and $\theta_{j, Moore's}^0$ are equivalent. The primary hypothesis, H_1 , indicates that the TFDEA technological index is closer to unity than that of Moore's law, and as such provides a closer estimate to the state of the art. Table 25 indicates that the TFDEA technology indices are closer to unity with a an over 99% probability than those of Moore's law with both 24 and 18 month periods.

Table 25 - Results of Null Hypothesis Test for DEA vs. Moore's Law

	<i>t</i>	<i>p</i>
24 Months	4.77750	0.00001
18 Months	4.177100	0.00011

7. Analysis of Results

The purpose of this case study was to apply the proposed methodology to a more complicated technology and expand it to multiple dimensions. The results of this study however were not as well defined as that of the two-dimensional example, but a number of lessons were learned.

a. Dataset Size

The small size of the dataset when combined with the large number of factors led to a larger number of processors being declared SOA. Although this is expected, this reduces the number of times that a SOA surface may move. To perform a better analysis, a larger set of data should be used than the 53 that were used in this example. In fact, there were a number of years where the number of new products was not twice the number of inputs and outputs. Future research could examine the impact of the number of products and the periodicity chosen.

b. Effectiveness of the Forecast

The method did work for setting the conservative estimates of the technology envelope. This provides a possible means to reexamine microprocessors in more detail as time passes.

c. Comparison with Other Methods

This case study was an expansion of Moore's Law. When directly compared to

Moore's Law, TFDEA provided a closer assessment to the SOA. By building a more comprehensive model multiple dimensions could be applied and as such allows for a more comprehensive picture.

C. Persistent Storage: Application to Multiple Technological Approaches

In the *Innovator's Dilemma*, Clayton Christensen [26] examines thirty years of data available through the DISK/TREND dataset. Over the span of that time a number of technological approaches to persistent storage were analyzed. A subset of this dataset is further analyzed. John Porter of DISK/TREND recorded the status of the hard disk and removable storage industry for nearly thirty years. This comprehensive collection of disk drive specifications included all hardware manufactured in that time with standard metrics to track the progress.

1. Determine Scope of Forecast

As time passed and the industry matured so did the information collected, and as such removable media and other elements were separated into their own studies. This study focuses on the years between 1994 and 1999, the last year that DISK/TREND data is available [82][86][85][84][83][87]. The scope of the forecast will be hard drive technology for this time period.

2. Define a Product

Contrary to Christensen's work where different form factor drives constituted different

technologies; a product for hard drive technology will include all form factors available in a single common model.

3. Define SOA Characteristics

Persistent storage technology requires a volume to contain an amount of information that one can recover. With increasingly expensive data center real estate and increasingly large database sizes, it is important to put as much data into as small of space as possible. In addition, smaller physical size may provide opportunities for lower production and distribution expenses. This model is presented in Figure 32.

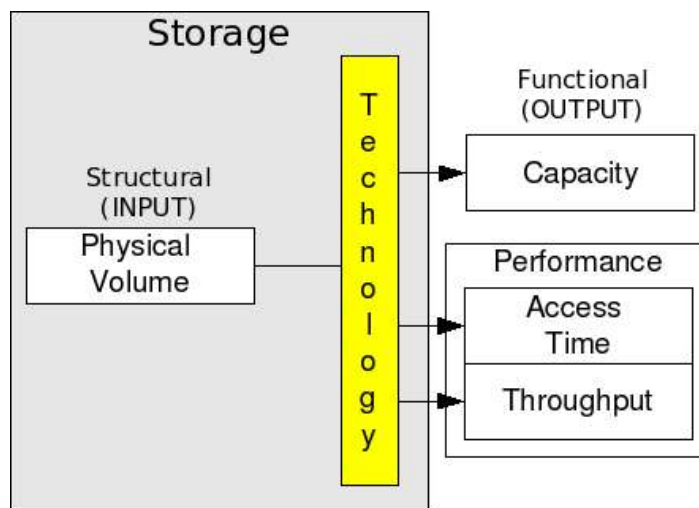


Figure 32 - Persistent Storage Functional and Structural Model

In the *Innovator's Dilemma*, Christensen notes that hard disk drive formats changed over the years resulting in a number of defunct companies which focused on technical approaches that were obsoleted by newer technologies. The twelve inch spindle fell to

the ten inch spindle which fell to the eight inch spindle and so forth. As time progressed, those companies that did not implement smaller spindles were replaced by companies that did. Although there was benefit to the larger devices, the reliability and the capacity of the smaller drives grew to a point that they were useful to the alternative markets. As physical volume decreased, this allowed more spindles in smaller spaces it was possible overall to have more storage in less space taking less data center room.

As for function, those requiring persistent storage are looking for a number of things. First is typically the capacity, or the amount of data that a device can hold. Over time the amount that can be stored by persistent storage has increased dramatically. In 2004, it is possible to purchase a 300 gigabyte drive for a \$246 [110] smaller than a paperback novel which would have required large cabinets less than ten years ago.

Capacity, however is not the only important characteristic. Two other characteristics which outline performance can be used. The rate at which data can be extracted and written to storage is very important. Many are familiar with the slow speed of USB when transferring photos from a digital camera to their computer. Examples of high throughput applications include restoration of lost data as well as streaming video on demand. When not dealing with large streams of continuous data, as is the case for many enterprise databases, the ability to access random points of data from the storage is also very important. This is measured by the average access time of a storage device, which

may be a composite of positioning and reading or writing operations. Lower access times permit many different small pieces of data to be accessed quickly. Much of the recent popularity of RAID devices has been driven by the need to access data more quickly than is physically possible from single spindles alone.

These three factors differentiate most storage from others. Tape devices allow for rapid streaming of large amounts of data but do not allow for rapid access of multiple records. Hard disk drives allow for rapid streaming of large amounts of data with relatively short access times. Solid state devices allow for quick data access and rapid random access because there are no physical limitations to the media being used to seek information, but limit the total amount of data which can be accessed. As time has passed, the usefulness of tape media has been called into question as the amount and data rates of disk storage has approached that of tape devices. In addition, the solid state devices today have approached and passed those of old disk and tape drive technology.

Traditional means of measuring these technologies have focused on the technical approach of the storage along the lines of rotational media, streaming media, or solid state media. In this model, the aim is to take all these into consideration by measuring the function provided by the volume.

4. Determine DEA Model

a. Orientation

While the overall goal of storage is to store more data in less space, manufacturers of storage media typically aim to increase the amount of storage, with only periodic changes in physical size or volume. For this reason the use of an output-oriented model has been chosen.

b. Returns To Scale

Due to physical requirements of the devices themselves, there is always an amount of volume that must be used, and a maximum size after which there are diminishing returns to scale. For this reason a VRS model has been chosen.

Since the output of access time is considered better when smaller, a transform has been used on the random access time as an inverse of the average time. The volume is transformed with a log function.

5. Collect Data

The data analyzed is a subset of the available data from Clayton Christensen due to the amount of data involved and the lack of complete data for years prior to 1994. For this reason the study focuses on disk drives present in the DISK/TREND reports between the years 1994 and 1999. The data itself reflects the rapid pace of the technology in the 1990's. In 1994, for example, only one measure of throughput was provided in the

specifications. By 1999, with the introduction of fibre channel and the pushing of the limits for a number of other standards, additional measures of throughputs were included. For this study, maximum throughput is the largest internal throughput published, or in the event that there is none provided, it is the maximum throughput. Since the publication date of the reports ranged from May-October in the years examined, only those devices listed as available in the first quarter of the year of the report were accepted as being released. The rest were checked to see if they were feasible, and then compared to future results to test the validity of the model. Although some products had the date listed to the month, only the quarter of release was used. For those which only listed the year of release, the second quarter was chosen so as to mitigate the impacts of large amounts of change.

6. Analyze Technological Progress

The data from DISK/TREND was entered as it was received in a year-to-year report. The results are displayed in Table 26, which reveals that the model is useful in mapping persistent storage technology. Over a quarter of the SOA models were predicted over the 10 years of data available. Additionally, the lower bound was predicted to be 32 percent, meaning that the conservative estimates were exceeded nearly 60% of the time. This can provide someone developing new products with a conservative estimate of the rates of technological change. Nearly half of the missed predictions were from 1986 and 1987

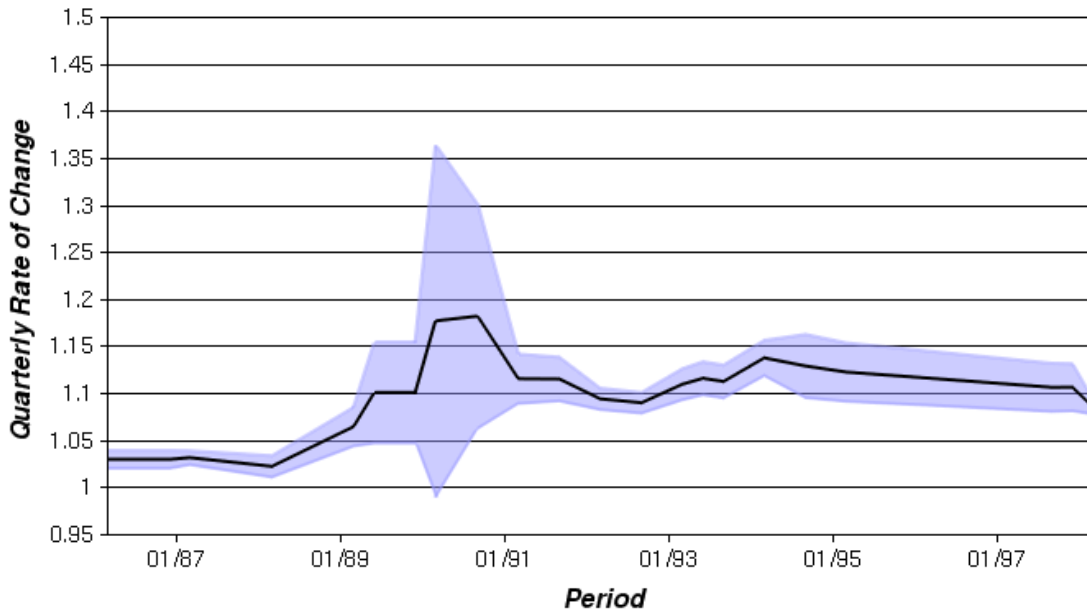
when the system was stabilizing accounting for 68 of the 141 missed forecasts.

Dropping these two quarters increases the percentage of accurate predictions to 34%.

Table 26 - Persistent Storage Validation Results with Output-Oriented Model

Predicted Range	119	26.21%
Above Conservative ROC Estimates	149	32.82%
Below Aggressive ROC Estimates	45	9.91%
Did not Predict	141	31.06%
Total	454	100.00%

Those predictions above the conservative forecasts drops to 31%. This means that 65% of the SOA products were above the conservative estimate for the SOA for 10 years. Examination of the ROC over time in Figure 33 reveals a point that corresponds to an introduction of a smaller form factor in 1991, which provides a large variance in the



created with ChartDirector from www.adusofteng.com

Figure 33 - DISK/TREND Rates of Change for Output-Oriented Model

rate of change. In the early years, this corresponds to a large number of the unpredicted SOA products. Once the primary effect of this has dissipated, the rates of change decrease. This model provides a means to assess future forecasts; however, it may be expanded to use diminishing returns to scale. In addition, it appears that the rate of change briefly accelerates and then stabilizes.

Further examination of the data and application of the IO-OO output-oriented model reveals a slight improvement in the forecasts. The data from the IO-OO model is illustrated in Table 27, which shows a marked improvement in predicted SOAs as well as a reduction in the overall number of missed predictions. Similar to the straight output-oriented model, a large portion of the missed forecasts are based on the 1986 and 1987 data, accounting for 62 of the missed predictions and only 1 accurate prediction as well as 62 lower bound predictions. Dropping these two years brings the combined lower bound prediction to nearly 75%.

Table 27 - Persistent Storage Validation Results with IO-OO Output-Oriented Model

Predicted Range	215	36.69%
Above Conservative ROC Estimates	183	30.10%
Below Aggressive ROC Estimates	112	19.11%
Did not Predict	76	12.97%
Total	586	100.00%

DISK/TREND data for each year was released during that year with forward looking statements concerning the release of disk products. Consequently, it is possible to

identify those drives that may have slipped their release date or restated their capabilities after the fact. To check this, all data was analyzed from the first quarter of the year of release and future releases were tested for feasibility. Interestingly, only two were above the upper bounds of the SOA, and both were restated in the subsequent year (1995). The IBM DCHC-38700 Ultrastar2 XP and DCMS-310800 Ultrastar2 XP were both scheduled for the first quarter of 1995. The IBM DCHC-38700 was never released, but the DCMS-310800 was released a year late, during the first quarter of 1996. These two cases illustrate the benefit of the tests to see if an unrealistic release date has been projected.

7. Analysis

The DISK/TREND case study's purpose is to examine a popular technology management forecasting literature study with the new methodology, and to further examine the model applied in multiple dimensions and technological approaches.

a. Identification of Unrealistic Expectations

The most notable changes in this study included the identification of two products which were too ambitious given the rate of technological change experienced at the point in time they were meant to be introduced. What this effectively means is that the method was an effective tool to identify overly aggressive performance goals which were never met. For the product development organization, this can provide the some foresight to avoid costly mistakes involving labor, resources, and image.

b. Expansion of Scope

Another notable result is the need for further model expansion. Although the double oriented model was used for validating the forecast, there were still a significant number of infeasible forecasts, or those that were out of the reasonable boundaries of the study. Of these a large portion occurred early in the study because the form factor changes were not foreseeable by the model as it stands. This issue provides a set of future research.

c. Identification of Disruptive Technologies

Identification of a new technology was present in the change of drive form factor; where a large amount of variance in technological progress was identified. This jump quickly evened out and became part of the model. Much of the early validation predictions were out of scope due to the jump in form factor.

d. Effectiveness of the Model

The effectiveness of the forecast was less than that of the TPC case study, but more than the expansion of Moore's Law. The double oriented model was able to forecast SOA storage 37% of the time, and evaluate the conservative estimate of the rate of change over 60% of the time. This indicates that the technique could be used to measure the conservative estimates of the technological rate of change. In addition, it should be noted that a significantly larger portion of data was available then for the other forecasts.

e. Comparison to Other Techniques

The change in form factor was addressed by Christensen by a completely separate technology forecast which forecasted the future of that form factor. When performing forecasts however, this leads to one of the primary tenants of the *Innovator's Dilemma*. By disregarding a technological approach, a forecaster or organization specifically disregards that which may be their doom. This method provides a means to analyze multiple technology approaches with the same method.

D. Analysis of Results

The three case studies were chosen to validate the proposed methodology, TFDEA, created by this research. Examination of these results reveals that the methodology is still sensitive to the effects of disruptive technology, but the effects overall are eventually absorbed by the system. TFDEA identified a disruptive technology by discovering a large rate of change required to deliver the cost performance announced by HP. Examination of this data revealed that the product was no longer using proprietary operating systems, and as such was able to do the job for significantly cheaper.

1. Effectiveness of the Model

The model was able to predict a large number of state of the art products. In providing conservative and aggressive estimates, the methodology was able to predict many future technologies. These issues can be related to the research questions below.

a. Using DEA to Represent SOA and Technology Trade-off Surfaces

Research question #1, deals with the use of DEA to measure trade-off surfaces and the state of the art is restated below.

<i>Research Question #1:</i>	<i>How can DEA be used to measure the SOA and trade-off surfaces?</i>
-------------------------------------	--

DEA proved to be an effective means to assess the state of technology. It was also able to handle the variable returns to scale. In the case of microprocessors it was able to better predict new product's then Moore's Law. It more accurately mapped the state of microprocessor technology as indicated by higher average technology indices than that of the single variable exponential model. This means that someone using this method to perform forecasts should get a better picture of what the technology has to offer. For persistent storage, TFDEA provided a method to represent multiple technological approaches on a single surface contrary to the work of Christensen. In his work different form factors were represented by different trending lines. By allowing all the technology to be represented on a single line, it becomes possible to monitor the change of technologies.

b. Extension of Temporal DEA to Monitor Technology

The second research question extended the first to monitor changing technology over time and is restated below.

<i>Research Question #2:</i>	<i>How can temporal DEA be extended to monitor technology of products that are only observed only once at irregular intervals?</i>
-------------------------------------	---

In the two dimensional model, it provided a more accurate means to assess the state of the art than basic regression through reduced residuals. This implies that it mapped closer to the realities of the technology than that of a simple regression model. In the second case however the dataset size was notably smaller than desired. This in turn reduced the effectiveness of the forecast in that there were not enough observations to build better frontiers. In the final case, TFDEA provided a means to measure the progress of technology through multiple generations of form factors. Although previously cited disruptive technologies, alternate form factors, did impact the forecasts, their impacts stabilized. This indicates that the model provides a means to properly assess and monitor the state of technology over time.

c. Extension of Temporal DEA to predict future technologies

The final research question guiding this research extended monitoring of past technologies to ascertain the characteristics of those in the future and is stated below.

<i>Research Question #3:</i>	<i>How can temporal DEA be used to forecast future attributes of technological products?</i>
-------------------------------------	---

TFDEA proved to provide insight into the forecasting of future technologies, however there are some issues that have arisen. In both TPC and persistent storage, double

orientation was required to expand the scope of the forecast, this indicates that choosing only one orientation for analysis is too limiting in scope. This is analogous to disregarding other technologies which may prove disruptive to the current ones. By extending the model to both output and input orientation, the number of SOA products which could be predicted were notably improved. For TPC, this handled all products, while for persistent storage it only handled a subset.

In addition, the model was resilient to previously defined disruptive technologies, or “spindle size.” Although a notable point of disruption was caused by reduced spindle size, the issue stabilized once a few observations had been made. This means that it provides a better means to assess the overall state of technology than the basic regression models of Christensen, through the multi-dimensionality of the problem.

2. Comparison to Other Techniques

The first example provided a straightforward example that was easily illustrated, and demonstrated its feasibility when compared to basic regression. There are a number of factors that permit the methods increased accuracy. The first issue is that of variable returns to scale. By allowing increasing and diminishing returns, TFDEA is able to better represent what the technology is capable of performing. Secondly, TFDEA provides a means to use both attributes, cost and performance, to more accurately represent the state of the technology over time.

For basic exponential trend extrapolation of Moore's Law, it offered the benefit of additional technology indicators. These technology indicators created a more comprehensive tool which could better represent the SOA.

The persistent storage example illustrated how multiple technology approaches can be measured with the same tool to address the dynamic nature of trade off surfaces. This provides a more thorough picture of the technology because it is able to monitor all of the technology together.

V. Conclusions

A. Research Overview

This research provides an overview of technology forecasting literature and discusses many tools and techniques technology forecasters use to assess and ascertain the capabilities of past, present, and future technologies. These methodologies are then narrowed to extrapolative techniques, or those that attempt to use historical data to measure past trends and estimate future technology capabilities. Such techniques are often used as exploratory methods to provide insight to poorly understood phenomena. These extrapolative techniques are then extended using the concepts of state of the art and trade-off surfaces.

Trade-off surfaces and state-of-the-art quantify and represent technology with a multidimensional surface. By doing so, a forecaster can approximate the nature and rates of technological change and use them to estimate future technology characteristics. Unfortunately, there are a number of outstanding gaps in the current literature for extrapolative techniques using state of the art and trade-off surfaces.

Although a number of options have been suggested such as ellipsoid projection, regression, and scoring models, each has major drawbacks. Ellipsoid projection, technometrics, and regression are central point methods that do not emphasize the “best” technologies, but target the average of available technologies. These methods attempt to

mitigate the effects of outliers when it is the outliers that a forecaster is most interested in predicting. When developing new state of the art products, organizations are not merely targeting something that is feasible, they want something that is the best. This is summarized by gap #1 below.

Gap #1:	<i>Current extrapolative forecasting methodologies do not address the “best” available technology but an aggregate of all technologies available.</i>
----------------	--

Furthermore, regression and scoring models do not take into account the trade-off shifts or fundamental structural change that occur as technologies evolve. This is summarized by gap #2.

Gap #2:	<i>Current extrapolative forecasting methodologies do not take into account the dynamic nature of trade-off surfaces.</i>
----------------	--

Expanding on gap #2, current technology forecasting techniques require characteristics to be non-correlated. When characteristics are correlated, current techniques either require composite characteristic indices which may or may not continue to be correlated over time. This is summarized by gap #3.

Gap #3:	<i>Current extrapolative forecasting methodologies require non-correlated attributes.</i>
----------------	--

The final gap with SOA techniques described by this research is that of the single output technology index which does not allow for disaggregation. Current

methodologies create a single fixed index that does not allow for disaggregation of information to better assess future capabilities of a technology. This is presented below as gap #4.

Gap #4:	<i>Current extrapolative methods are limited to a single output that can not be disaggregated.</i>
----------------	---

DEA offers a way to address many extrapolative technology forecasting gaps because DEA's inherent strengths directly correspond to current technology forecasting technique weaknesses. As an extreme point method, it measures outliers, but is not adversely affected by products which are not SOA. It can assess multiple inputs and outputs simultaneously, giving results that can be disaggregated. DEA can also address variable trade-offs because each DMU effectively chooses what trade-offs best represent themselves. The shortcoming with with DEA is that there is no method to measure the movement of a DEA frontier without multiple observations of a DMU over regular intervals, which is presented as gap #5.

Gap #5:	<i>Current temporal models of DEA do not allow for DMUs which are observed at only one time period and introduced at irregular intervals.</i>
----------------	--

Based on these five gaps the research objective was synthesized and is summarized below.

Research Objective:	<i>Develop a methodology for technology forecasting which provides for a readily calculable method to measure the SOA and its advance by extending current temporal DEA to allow for DMUs which are introduced once at irregular intervals.</i>
----------------------------	--

To achieve the research objective, a set of research questions are developed, and a methodology is created to address them. The first question is designed to determine how DEA can be used to represent the current technological SOA and is summarized below.

Research Question #1:	<i>How can DEA be used to measure the SOA and trade-off surfaces?</i>
------------------------------	--

DEA can be used to measure the SOA and trade-off surfaces by creating a structural-functional model of a technology and using products implementing that technology to create the trade-off surface. By doing this, all technology is compared to a set of “best” technologies in the form of an index that can be disaggregated. Once a surface is constructed, it is necessary to determine how that surface changes over time. This is addressed by research question #2, stated below.

Research Question #2:	<i>How can temporal DEA be extended to monitor technology of products that are only observed only once at irregular intervals?</i>
------------------------------	---

Changes in DEA relative efficiencies over time can be used to calculate the technological rate of change over time. By averaging out the change of technological indices over time, a fixed rate of change can be calculated. With the current SOA constructed, and its past rate of change determined, the remaining issue for the

technology forecaster is to determine future technology characteristics resulting in research question #3 summarized below.

<i>Research Question #3:</i>	<i>How can temporal DEA be used to forecast future attributes of technological products?</i>
-------------------------------------	---

Future capabilities of technology can be assessed in two ways. The first is to project the current SOA to the future and examine the feasibility of a proposed future technology. The second method is to compare the future product characteristics with the current SOA and use the present ROC to see if it is a feasible product based on the announced time of introduction and the calculated results.

The interrelationships of the gaps, objective, and research questions are included in Figure 17 which is redisplayed as Figure 34. Validation of the new methodology is provided by three case studies. The first case study examines OLTP database systems and was chosen to provide a straightforward easily illustrated two dimensional example to demonstrate the methodology's applicability. The second case reexamines Moore's law and extends it to six dimensions. It was chosen to examine its feasibility in forecasting more complex technologies by extending the model to multiple dimensions. The third case study, persistent storage, was chosen to validate the model when applied to multiple technological approaches.

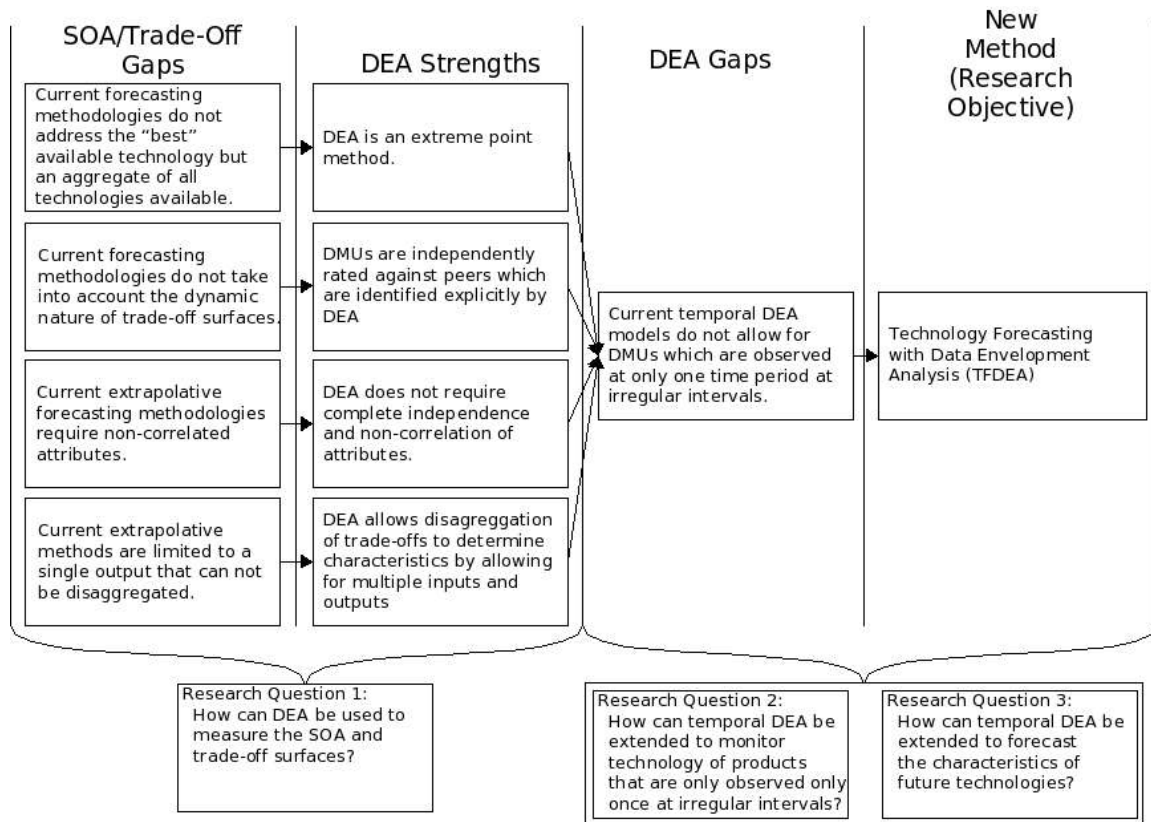


Figure 34 - Research Design

B. Results of Case Studies

Each of the case studies chosen provided insight into the feasibility of TFDEA in monitoring and forecasting the future capabilities of technologies. These results are summarized below.

1. OLTP Databases

For the first case study, OLTP databases, TFDEA was able to use historical data projected into the future to accurately predict a large percentage of the future SOA products. TFDEA also identified a new technological approach as being disruptive by identifying the sudden drop in price/performance with the introduction of Linux to large

scale OLTP systems.

When compared to standard regression analysis, TFDEA was more effective at predicting future technologies. TFDEA significantly reduces the residuals between forecasted and actual product release dates. The OLTP case study also illustrated the need to take into account the need to address both input and output orientation simultaneously due to ever decreasing costs of entry in an otherwise output-oriented technology. When lower costs were introduced, the output-oriented model breaks down due to the nature of VRS and super efficiency with DEA.

2. Moore's Law Reexamined

The second case study reexamined Moore's Law and extended it to six dimensions. TFDEA provided a more accurate representation of the future than using Moore's Law to approximate a future technological index. The case study demonstrated the need for an adequate amount of historical data to forecast future technology characteristics. In this case study, the low number of both product introductions per period and number of periods did not provide adequate time for the model to stabilize. Consequently, the conservative rate of change estimates provided a low mark that all subsequent SOA processors exceeded. However, most forecasts exceeded the aggressive estimates as well indicating that additional data or model modification may be required.

3. Persistent Storage Reexamined

The final case study, persistent storage, demonstrated TFDEA's capability to assess multiple technological approaches with a single forecasting tool. In his book, *Innovator's Dilemma*, Clayton Christenson assesses multiple form factors of hard drives as different technologies [26]. Each technology is represented by a separate trend line, and those trend lines are used to assess an individual technologies. Using TFDEA, all technologies can be assessed in the same model, focusing more on the function of the technology rather than the form used to provide that function.. Although not as accurate as the OLTP study, TFDEA successfully identified two technologies as being infeasible prior to release.

This case study illustrates a shortcoming that remains with TFDEA in that when validating forecasts there are occasional infeasible solutions to the linear program. The occurrence of infeasible solutions was decreased by using both the input and output-oriented rates of change to expand the SOA frontier. While this approach solved the problem of infeasible solutions in the OLTP case study, it did not completely solve the problem for the persistent storage case study.

4. Discussion of Results

As proposed, TFDEA uses the mean of the ROC mean to predict future technology characteristics, and therefore imposes a stricter assessment of the frontier than may be

required. Because of this, it is not unexpected that forecast accuracy would be less than 95%. Although many new products fall outside the forecasted confidence intervals, TFDEA does reliably provide conservative ROC estimates. Therefore the method can be used to assess the conservative limits of technologies being developed. This is valuable.

One expected strength of this method was resistance to disruptive technologies. The persistent storage results demonstrated that there is still sensitivity to disruptive technologies. Since forecasts can only be constructed from currently produced sets of inputs or outputs, it is difficult to accurately predict breakthrough products. In DEA, accuracy may be improved by using alternative transforms, scale efficiency, or slack efficiency over time. Another potential solution may be to use the multiplicative DEA model. This is beyond the goals set forth in this research, but could be targeted for future research.

Interestingly, methods such as the IO-OO model do expand the breadth of the forecast, but do not increase the accuracy of the forecasts. A better way may be to validate against both orientations in the event that the technology being studied is shifting its orientation from output to input.

This change in orientation concept can be explained by considering the nature of a specific technology. Throughout the 1990s, the demands of office productivity software pushed personal computer hardware technology performance requirements to ever

increasing levels. However, more recently, the demand for ever increasing performance have given way to the demand for more affordable computers. In other words, the focus of technology developers have shifted from output-orientation to input-orientation.

One possible way to assess an industry's state is to analyze the percentage of products that are technically infeasible when they are announced. In cases where the output-oriented model produce more infeasible products, the technology may be more mature. Thus the primary goal of manufactures is to produce relatively constant outputs with fewer resources. On the other hand if there are more infeasible products when using input-orientation, then manufactures are attempting to push the technology to higher levels of performance.

C. Research Contributions

This research provides a new methodology for tracking technological progress and predicting the technology characteristics that was not previously available. Table 28 provides a summary table of the contributions followed by an explanation.

Table 28 - Research Contributions

<i>Contribution</i>	<i>Research Question Addressed</i>
Method using DEA to represent the SOA	Research Question #1
Method for monitoring SOA advancement	Research Question #2
Method for forecasting future SOA	Research Question #3
Linking DEA and Technology Forecasting	--

1. Method Using DEA to Represent the SOA

Gap #1 addresses the fact that current methods used to represent trade-off surfaces do not address the “best” technology available, they provide a weighted or averaged aggregate of all available technology and intentionally disregard outliers. When non-state of the art products are released using current methods, all measurements of the state of the art are affected. TFDEA avoids this issue by only using the best available technologies. This provides forecasters an estimate of future technology limits. Scoring models may provide specific functional constraints, but they assume fixed trade-offs throughout time. For technology, this is not always the case. TFDEA avoids this problem by allowing for multiple sets of technology trade-offs simultaneously.

One of the more difficult aspects of technology forecasting is the identification of pertinent attributes. Gap #2 complicates this because many current methods rely on inflexible indices which are locked into *a priori* weighting schemes. For those methods which do not assume *a priori* weights, this method provides an applicable and executable procedure which was not previously available to practitioners through extrapolative means. Gaps #2 and #3 further complicate this identification due to limitations of current methodologies. Characteristics are often correlated which can result in unreliable trade-offs for correlated attributes using linear regression. Since DEA is less sensitive to correlated measures, it offers a means to analyze technological change even when

interrelationships between measures exist.

Gap #4 discussed in the current SOA and trade-off surface literature is that the single technology index is fixed and difficult to disaggregate. Regression and scoring models, although able to handle a large number of inputs, cannot handle a large number of outputs simultaneously. Although DEA is in itself an aggregate score, it does not suffer the drawbacks of other methods. It allows for each input and output to be decoupled from one another thus not requiring the assumption of fixed interrelationships. One of the great strengths of TFDEA is its ability to tune both the form and function of the technology to determine the best relative performance to all peers.

2. Method for Monitoring SOA Advancement

Gap #5 in the literature is partially addressed by analyzing only the shift in the frontier and not the relative position of each DMU for DEA. TFDEA does this by recording those products which define the SOA at a point in time and assessing their change in relative efficiency over time. By averaging this change, TFDEA allows DEA to be extended to answer Research Questions #2. Prior to this research there was no methodology for temporal DEA that involved multiple DMUs observed only once introduced at varying time periods.

3. Method to Forecast Future SOA

Gap #5 in the literature is further addressed by radial projection of a given SOA

surface into the future by using the ROC determined by the research answer #2. In addition, the feasibility of future product characteristics can be assessed by using the TFDEA predetermined ROC and SOA surface to identify the expected time of release. Those applications that cannot use methods established by Malmquist Productivity Indices can look to TFDEA as an alternative.

4. Linking DEA to Technology Forecasting

With over 1500 journal citations to its credit, DEA offers the potential to leverage a large amount of research to better address technology forecasting issues. This initial research provides the first direct mapping of technology forecasting to DEA and identifies a number of potential areas of future research in subsequent sections. Many of the questions or issues identified in this study may benefit from other DEA research. With such a rich set of historical applications and variations, TFDEA offers a beginning to a more detailed set of research to overcome some of the identified shortcomings of the present methods and provides a means to address shifting efficiency scores of an established method, which had not previously possible. Although not perfect, the method does provide insight into the future limitations and identification of areas which can be expanded upon by an expansive set of other applications and permutations offered by DEA.

D. Management Implications

The current tools for technology forecasting offer a limited set of tools to quantitatively forecast the future of characteristics of technology based on historical data. This research provides a methodology for the technology forecaster to better estimate the future characteristics of technologies.

1. Requirements

There are a number of requirements that must be met in order to implement TFDEA.

a. Data

There are a number of data requirements that the forecaster must address when implementing the methodology.

Discrete Data

Since DEA is an extreme point method requiring the construction of a frontier from well defined data, it is necessary for the data to have discrete well defined values.

Adequate Sample Size

Another important issue that must be considered is that of the sample size. An adequately large number of products (DMUs) are necessary to construct a viable efficiency frontier. A general rule is that the number of products (DMUs) should be three times the number of inputs and outputs.

Adequate Number of Periods

In addition to the adequate number of observations per period, an adequate past history is required. This means that only those technologies which have existed for a number of periods should be analyzed.

b. Understanding of Technology Structure

Although not as dependent on understanding as other methods, a rudimentary understanding of the technology being forecast is required to use TFDEA. The better the understanding of the pertinent structural and functional characteristics, the broader the applicability of the technique can be.

2. Strengths

a. Focusing on Best Technology

Technology forecasters frequently try to determine the highest feasible levels of technology that will be available at future points in time. The SOA as defined by Dodson, relies on outliers that central point methods attempt to disregard. The *Innovator's Dilemma* indicates that an organization must decide whether or not to disregard technical approaches that may render their current products obsolete. By using outliers to forecast future characteristics, management can better assess the future state of technology.

b. Ability to Forecast Future Characteristics

An advantage of TFDEA over scoring and regression is the ability to forecast future characteristics using the ROC and the characteristics of a given portion of the SOA frontier. This allows forecasters to estimate potential future product characteristics, regardless of trade-offs.

c. Ability to Estimate the Availability of Future Characteristics

A corollary to predicting future characteristics, is the ability to determine the time a product with given characteristics will be made available. Despite the trade-offs for a given technical approach, a ROC and SOA frontier make it possible to predict when a product will be available.

d. Ability to Evaluate Multiple Technical Approaches Simultaneously

Although previously indirectly stated, one TFDEA's greatest strengths is its ability to simultaneously assess multiple technical approaches. This partially avoids the *Innovator's Dilemma*, because it does not allow a forecaster to ignore alternative technologies.

3. Limitations

a. Sensitivity to Disruptive Technologies

Although more resistant to disruptive technologies due to the strengths listed above, TFDEA can still be sensitive to disruptive technologies. When a technology experiences

great amounts of change in trade-offs, TFDEA may not anticipate them. However, TFDEA is capable of incorporating those changes into future forecasts .

b. Assumption of Constant Rates of Change

A major limitation of this research is the assumption of constant rate of technological change. There are likely alternative methods which may better describe the rate of technological change over time.

4. Summary

For organizations that rely heavily on technology, or those that are responsible for developing future products in competitive markets, TFDEA offers management a means to better predict the feasibility of future technological developments. In the third case study, two disk drives dates of availability identified as overly technologically aggressive by TFDEA were proved to be so. Although a small example, the implications are significant for decision makers. Basing growth or organizational changes on technologies that may not appear in the announced time frame can result in costly mistakes.

For development teams that seek to provide cutting edge technology ahead of their competitors, TFDEA provides a way of setting realistic targets or identifying where they expect the competition to be. Like all forecasting techniques, this technique should be used in conjunction with other intelligence to assess the feasibility of any new products

which seem to violate past trends.

For marketing organizations working with development organizations that fail to achieve predicted trends, this offers a way to target unidentified markets, or a illustrate early differentiation from the competition. When combined with market intelligence, TFDEA would help an organization to identify itself as a front runner, and assessing the risk of eminent competition. Due to the nature of DEA, it may also be possible to use DEA weights to identify neglected markets or new ways to market the product through increasingly targeted marketing.

The steps involved with TFDEA are straight forward. First, determine the scope of the technology to be forecast. Second define what a product of the technology is. Third define those structural characteristics that are required for the technology to deliver a set of functional characteristics. Fourth, based on the nature of technology the proper DEA model is determined. Then data is collected, analysis is performed and the appropriate judgments are made. Once an acceptable model is created then the results of the model can be used to predict the characteristics of future technologies.

E. Future Research

Although the research has provided a new and useful tool for the technology forecaster, there are a number of future points of research that can be further performed to expand the tool and make it more useful.

1. Related to Data Envelopment Analysis

The method presented herein provides a number of potential extensions to DEA research. Two potential extensions of the method are suggested below.

a. Identification of Sample Size Requirements

An aspect of DEA that has received little attention is how to identify the minimum amount of data required to provide meaningful results. As a general rule, three times the number of inputs and outputs combined are suggested. However, the nature of technology forecasting further complicates this issue. Since frontiers are defined multiple times, a larger set of data is required. This may mean that the amount of data required decreases as time passes or once an adequate frontier is established, but at current it is not known, and as such requires further investigation.

b. Scale Efficiency

DEA has ways to assess the distance between VRS and CRS frontiers via scale efficiency. This could provide a methodology to more accurately portray technological rates of change based on a products location on the efficiency frontier. Those near VRS points of inflection would be expected to make less progress than those on the fringe, while those which were previously considered infeasible may be better taken into account for by using scale efficiency to explain what the rate of change is.

c. Variable Rates of Change

The model chosen here was that of a constant multiplier. It has the ability to map technological progress over time based on the assumption of a constant rate of change. Additional methods should be used to ascertain if they better assess the rate of technological change.

d. Analysis of the Rate of Change

This study, assumes that the ROC is a normal distribution. Analysis of this would provide more information and a more accurate means of forecasting if it is not true.

2. Related to Technology

Technology itself can be further examined based on the nature and the output of the analysis. By examining the results of the analysis, it may be possible to get additional insights from the analysis.

a. Examination and Comparison of Orientation

Comparison of input versus output-orientated DEA models may provide useful insights. For technologies that are focussed on greater and greater performance, output-oriented models are ideal. This can be thought of as the growth period for a specific technology. For processors and storage devices, Moore's Law is driven by the ever present demand for more power at faster speeds. Within more mature industries such as transportation, the emphasis is on providing the same function for less input. Reducing

moving parts, or the other elements to deliver similar performance is the intent of new technological developments. In these cases input-orientated models may prove a better means to predict future technologies.

More importantly, there may be a time when such switchover in orientation is visible. In these cases, TFDEA may provide an indicator for maturing industries to change their strategy for future products. Understanding where this tipping point occurs may provide a competitive advantage.

b. Further Studies with Additional Technologies

As with any new methodology, additional studies using DEA for technological forecasting are needed. The model can provide a way to further study changing efficiency. In addition, by no longer relying on multiple observations of the same DMU, it offers an extension to traditional DMUs.

c. Integration with Technological Approach Forecasting

By forecasting individual technical approaches, it may be possible to fit them with the areas of the SOA that apply to them or identify possible future targets for development which may or may not have been considered.

d. Integration with Expert Opinion

Perhaps one of the most interesting areas of potential research is for a combination of the technique with methodologies used to identify future technical approaches. By

taking into account the entire spectrum of a product, it may be possible for companies to better identify potential technical approaches to push multiple market boundaries at once.

e. Integration with Preference Structure

Zhu and Seiford have already introduced means to identify best practices and user preference. This could be used to target future market technologies or identify potential new markets for current and developing products.

f. Usage of Multiplier Model to Identify Primary Technology Drivers

Through the use of the DEA multiplier model, identification of key technology drivers through the attribute weights can help identify the potential for future development as well as identification and filtration of important attributes.

3. Identifying Future Markets or New Targets

Analysis of the frontier in order to identify future targets or segments not addressed may be a means for a product manager within an organization to utilize TFDEA as a means to set new product targets. Not only does this provide the implementor a means to identify obtainable goals given the current environment, but also a means for the end-user to identify possible new applications based on the product's peers.

References

- [1] A. J. Alexander, J. R. Nelson, "Measuring Technological Change: Aircraft Turbine Engines," *Technological Forecasting and Social Change*, vol. 5, pp. 189-203, 1973.
- [2] A. J. Alexander, B. M. Mitchell, "Measuring the Technological Change of Heterogeneous Products," *Technological Forecasting and Social Change*, vol. 27, pp. 161-195, 1985.
- [3] R. Allen, A. Athanassopoulos, R. G. Dyson, E. Thanassoulis, "Weights restrictions and value judgments in data envelopment analysis: Evolution, development and future directions," *Annals of Operations Research*, vol. 73, pp. 13-34, 1997.
- [4] P. Andersen, N. C. Petersen, "A Procedure for ranking efficient units in data envelopment analysis," *Management Science*, vol. 39, pp. 1261-4, 1993.
- [5] T. R. Anderson, S. Grosskopf, R. Fare, X. Song, *Examining Moore's Law Using Data Envelopment Analysis*, 2001.
- [6] T. R. Anderson, "Data Envelopment Analysis," *Encyclopedia of Information Systems*, : CRC, 2002.
- [7] T. R. Anderson, R. Fare, S. Grosskopf, L. Inman, X. Song, "Further Examination of Moore's Law Using Data Envelopment Analysis," *Technological Forecasting and Social Change*, vol. 69, pp. 465-477, 2002.
- [8] R. U. Ayres, *Technological Forecasting and Long-Range Planning*, New York: 1969.
- [9] R. U. Ayres, "Empirical Measures of Technological Change at the Sectoral

- Level," *Technological Forecasting and Social Change*, vol. 27, pp. 229-247, 1985.
- [10] R. U. Ayres, "What Have We Learned," *Journal of Technology Forecasting and Social Change*, vol. 62, pp. 9-12, 1999.
- [11] R. C. Baker, S. Talluri, "A closer look at the use of data envelopment analysis for technology selection," *Computers & Industrial Engineering*, vol. 32, pp. 101-108, 1997.
- [12] R. D. Banker, A. Charnes, W. W. Cooper, J. Swarts, D. Thomas, "An Introduction to data envelopment analysis with some of its models and their uses," *Research in Governmental and Nonprofit Accounting*, : JAI Press, 1989.
- [13] , *Frontier Analysis*, 1996.
- [14] F. Betz, *Strategic Technology Management*, New York: 1993.
- [15] G. E. Box, G. M. Jenkins, G. C. Reinsel, *Time Series Analysis Forecasting and Control*, Upper Saddle River, New Jersey: 1994.
- [16] J. R. Bright, *A Brief Introduction to Technology Forecasting*, Austin, Texas: 1972.
- [17] P. J. Brockwell, R. A. Davis, *Introduction to Time Series and Forecasting*, New York: 1996.
- [18] S. Bulla, W. W. Cooper, D. Wilson, K. S. Park, "Evaluating efficiencies of turbofan jet engines: a data envelopment analysis approach," *Journal of Propulsion and Power*, vol. 16, pp. 9, 2000.
- [19] A. K. Chakravarti, "Modified Delphi Methodology for Technology Forecasting,"

Technological Forecasting and Social Change, vol. 58, pp. 155-165, 1998.

- [20] A. Charnes, W. W. Cooper, E. Rhodes, "Measuring the efficiency of decision making units," *European Journal of Operational Research*, vol. 2, pp. 429-444, 1978.
- [21] A. Charnes, W. W. Cooper, "Managerial economics: past, present and future," *Journal of Enterprise Management*, vol. 1, pp. 5-23, 1978.
- [22] A. Charnes, W. W. Cooper, Z. M. Huang, D. B. Sun, "Polyhedral cone-ratio DEA models with an illustrative application to large commercial banking," *Journal of Econometrics*, vol. , pp. , 1990.
- [23] A. Charnes, A. Lewin, L. M. Seiford, *From Efficiency Calculations to A New Approach for Organizing and Analyzing : DEA Fifteen Years Later*, 1997.
- [24] T.-Y. Chen, T.-L. Yeh, "A Measurement of Bank Efficiency, Ownership Productivity Changes in Taiwan," *The Services Industry Journal*, vol. 20, pp. 95-109, 2000.
- [25] R. K. Chisolm G. R. Whitaker, Jr., *Forecasting Methods*, Homewood, IL: 1971.
- [26] C. M. Christensen, *The Innovator's Dilemma*, Boston, Massachusetts: 2000.
- [27] J. F. Coates, V. T. Coates, J. Jarratt, L. Heinz, *Issues Management*, Mt. Airy, MD: 1986.
- [28] W. W. Cooper, L. M. Seiford, K. Tone, "Book selection - data envelopment analysis: a comprehensive text with models, applications, references and DEA-Solver software," *The Journal of the Operational Research Society*, vol. 52, pp. 2, 2001.

- [29] W. W. Cooper, L. M. Seiford, K. Tone, *Data envelopment analysis*, 2000.
- [30] A. T. Court, "Hedonic Price Indexes with Automotive Examples," *The Dynamics of Automobile Demand*, New York: General Motors Corporation, 1939.
- [31] E. N. Dodson, "A General Approach to the Measurement of State of the Art and Technical Advance," *Technology Forecasting*, vol. 1, pp. 391-408, 1970.
- [32] E. N. Dodson, "Measurement of State of the Art and Technological Advance," *Technological Forecasting and Social Change*, vol. 27, pp. 129-146, 1985.
- [33] J. R. Doyle, R. H. Green, "Strategic choice and data envelopment analysis: comparing computers across many attributes," *Journal of Information Technology*, vol. 9, pp. 61-69, 1994.
- [34] J. R. Doyle, R. H. Green, "Comparing products using data envelopment analysis," *Omega*, vol. 19, pp. 631-638, 1991.
- [35] K. J. Edwards, T. J. Gordon, "Further Research into a Convention for Measuring the State-of-the-Art of Product," *Technological Forecasting and Social Change*, vol. 24, pp. 153-175, 1983.
- [36] C. W. Emory, *Business Research Methods*, Homewood, IL: 1976.
- [37] E. Esposito, M. Mastroianni, "Technological Evolution of Personal Computers and Market Implications," *Technological Forecasting and Social Change*, vol. 59, pp. 235-254, 1998.
- [38] R. Fare, S. Grosskopf, *Intertemporal Production Frontiers: With Dynamic DEA*, Boston: 1996.
- [39] R. Fare, S. Grosskopf, "Estimation of returns to scale using data envelopment

- analysis: A comment," *European Journal of Operational Research*, vol. 79, pp. 379-382, 1994.
- [40] R. Fare, S. Grosskopf, "Productivity Developments in Swedish Hospitals: A Malmquist Approach," *Data Envelopment Analysis: Theory, Methodology and Applications*, Boston: Kluwer Academic Publishers, pp. , 1994.
- [41] R. Fare, S. Grosskopf, B. Lindgren, P. Roos, "Productivity changes in Swedish pharmacies 1980- 1989: A non- parametric malmquist approach.," *The Journal of Productivity Analysis*, vol. 3, pp. 85- 101, 1992.
- [42] R. Fare, S. Grosskopf, P. Roos, "Malmquist Productivity Indexes: A Survey of Theory and Practice," *Index Numbers: Essays in Honour of Sten Malmquist*, Boston: Kluwer Academic Publishers, 1998.
- [43] M. J. Farrell, "The measurement of productive efficiency," *Journal of the Royal Statistical Society*, vol. 120, pp. 253-281, 1957.
- [44] B. Golany, "A note on including ordinal relations among multipliers in data envelopment analysis," *Management Science*, vol. 34, pp. 1029-33, 1988.
- [45] T. J. Gordon, T. R. Munson, "A Proposed Convention for Measuring the State of the Art Products or Processes," *Technological Forecasting and Social Change*, vol. 20, pp. 1-26, 1981.
- [46] T. J. Gordon, T. R. Munson, "A Proposed Convention for Measuring the State of the Art Products or Processes," *Technological Forecasting and Social Change*, vol. 20, pp. 1-26, 1981.
- [47] Z. Griliches, , Cambridge, MA: 1971.
- [48] S. Grosskopf, "Some Remarks on Productivity and its Decompositions," *Journal*

of Productivity Analysis, vol. 20, pp. 459-474, 2003.

- [49] W. E. Halal, M. D. Kull, A. Leffman, "The George Washington University Forecast of Emerging Technologies - A Continuou," *Technological Forecasting and Social Change*, vol. 59, pp. 89-110, 1998.
- [50] D. V. Hall, *Microprocessors and Interfacing: Programming and Hardware*, New York: 1990.
- [51] R. L. Hamblin, R. B. Jacobsen, J. L. Miller, , : 1973.
- [52] O. Helmer, *Looking Forward-A Guide to Futures Research*, Beverly Hills, CA: 1983.
- [53] J. L. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach*, San Francisco: 1996.
- [54] R. R. Hocking, *Methods and Applications of Linear Models - Regression and the Analysis of Variance*, New York: 1996.
- [55] Intel Corporation, Intel Processor Quick Reference Guide, 2004.
- [56] S. M. Kang, Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, New York: 1999.
- [57] A. Kayal, "Measuring the Pace of Technological Progress: Implications for Technological For," *Technological Forecasting and Social Change*, vol. 60, pp. 237-245, 1999.
- [58] M. Khouja, "The Use of Data Envelopment Analysis for Technology Selection," *Computers and Industrial Engineering*, vol. 28, pp. 123-132, 1995.
- [59] R. E. Klitgaard, "Measuring Technological Change: Comments on a Proposed

- Methodology," *Technological Forecasting and Social Change*, vol. 6, pp. 437-440, 1974.
- [60] K. E. Knight, *A Study of Technological Innovation -- The Evolution of Digital Computers*, 1963.
- [61] K. E. Knight, "A Functional and Structural Measurement of Technology," *Technological Forecasting and Social Change*, vol. 27, pp. 107-127, 1985.
- [62] H. A. Linstone, "TFSC: 1969-1999," *Technological Forecasting and Social Change*, vol. 62, pp. 1-8, 1999.
- [63] H. A. Linstone, M. Turoff, *The Delphi Method: Techniques and Applications*, Reading, Massachusetts: 1975.
- [64] A. Lipinski, D. Loveridge, "Institute for the Future's Study of the UK, 1978-1995," *Futures*, vol. 14, pp. 205-239, 1982.
- [65] H. Majer, "Technology Measurement: The Functional Approach," *Technological Forecasting and Social Change*, vol. 27, pp. 335-351, 1995.
- [66] S. Malmquist, "Index Numbers and Indifference Surfaces," *Trabajos de Estadística*, vol. 4, pp. 209-242, 1953.
- [67] J. P. Martino, "Measurement of Technology Using Trade-Off Surfaces," *Technological Forecasting and Social Change*, vol. 27, pp. , 1985.
- [68] J. P. Martino, "A Comparison of Two Composite Measures of Technology," *Technological Forecasting and Social Change*, vol. 44, pp. 147-159, 1993.
- [69] J. P. Martino, "Technological Forecasting for the Chemical Process Industries," *Chemical Engineering*, vol. , pp. 54-62, 1971.

- [70] J. P. Martino, *Technological Forecasting for Decision Making*, New York: 1992.
- [71] D. C. Montgomery, E. A. Peck, *Introduction to Linear Regression Analysis*, New York: 1982.
- [72] G. E. Moore, "Intel-memories and the microprocessor," *Daedalus*, vol. 125, pp. 55-80, 1996.
- [73] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. , pp. 114-117, 1965.
- [74] F. Narin, M. P. Carpenter, P. Woolf, "Technological Performance Assessments Based on Patents and Patent Citations," *Transactions on Engineering Management*, vol. EM-31, pp. 172-183, 1984.
- [75] F. Narin, D. Olivastro, K. A. Stevens, "Bibliometrics/Theory, Practice, and Problems," *Evaluation Review*, vol. 18, pp. 65-76, 1994.
- [76] R. C. Nyhan, L. L. Martin, "Comparative performance measurement: A primer on data envelopment analysis," *Public Productivity & Management Review*, vol. 22, pp. 348, 1999.
- [77] O. B. Oleson, N. C. Petersen, GAMS for DEA, 1994.
- [78] A. L. Porter, F. A. Rossini, "Technological Forecasting," *Encyclopedia of Systems and Control*, Oxford: Permagon, pp. 4823-4828, 1987.
- [79] A. L. Porter, M. J. Detampel, "Technology Opportunities Analysis," *Technological Forecasting and Social Change*, vol. 49, pp. 237-255, 1995.
- [80] A. L. Porter, M. J. Detampel, "Tech Forecasting an Empirical Perspective," *Technological Forecasting and Social Change*, vol. 62, pp. 19-28, 1999.

- [81] A. L. Porter, A.T. Roper, W. Mason, F. A. Rossini, J. Banks, *Forecasting and Management of Technology*, New York: 1991.
- [82] J. N. Porter, R. H. Katzive, 1994 DISK/TREND Report, 1994.
- [83] J. N. Porter, R. H. Katzive, 1995 DISK/TREND Report, 1995.
- [84] J. N. Porter, R. H. Katzive, 1996 DISK/TREND Report, 1996.
- [85] J. N. Porter, R. H. Katzive, 1997 DISK/TREND Report, 1997.
- [86] J. N. Porter, R. H. Katzive, 1998 DISK/TREND Report, 1998.
- [87] J. N. Porter, R. H. Katzive, 1999 DISK/TREND Report, 1999.
- [88] H. R. Rao, B. D. Lynch, "Hedonic Price Analysis of Workstation Attributes," *Communications of the ACM*, vol. 36, pp. 95-102, 1993.
- [89] D. Sahal, "Foundations of Technometrics," *Technological Forecasting and Social Change*, vol. 27, pp. 1-37, 1984.
- [90] D. Sahal, "On the Conception and Measurement of Trade-Off in Engineering Systems: A Case St," *Technological Forecasting and Social Change*, vol. 8, pp. 371-384, 1976.
- [91] D. Sahal, "The Generalized Distance Measures of Technology," *Technological Forecasting and Social Change*, vol. 9, pp. 289-300, 1976.
- [92] D. Sahal, *Patterns of Technological Innovation*, : 1981.
- [93] P. P. Saviotti, "An Approach to the Measurement of Technology Based on the Hedonic Price Method a," *Technological Forecasting and Social Change*, vol. 27, pp. 309-334, 1985.
- [94] L. M. Seiford, "A Bibliography for Data Envelopment Analysis (1978 - 1996),"

Annals of Operations Research, vol. , pp. 393-438, 1997.

- [95] L. M. Seiford, J. Zhu, "Infeasibility of super-efficiency data envelopment analysis models," *INFOR*, vol. 37, pp. 174-187, 1999.
- [96] L. W. Seiford, J. Zhu, "Sensitivity and stability of the classifications of returns to scale in data envelopment analysis," *Journal of Productivity Analysis*, vol. 12, pp. 21, 1999.
- [97] K. Shanley, History and Overview of the TPC, 2002.
- [98] R. W. Shepard, *Cost and Production Functions*, Princeton, NJ: 1953.
- [99] Standard Performance Evaluation Corporation, www.spec.org, 2004.
- [100] D. T. Taylor, R. G. Thompson, "DEA best practice assesses relative efficiency, profitability," *Oil & Gas Journal*, vol. 93, pp. 60-70, 1995.
- [101] S. Thore, F. Phillips, T. W. Ruefli, P. Yue, "DEA and the management of the product cycle: The U.S. computer industry," *Computers & Operations Research*, vol. 23, pp. 341-356, 1996.
- [102] K. Tone, "A slacks-based measure of super-efficiency in data envelopment analysis," *European Journal of Operational Research*, vol. 143, pp. 10, 2002.
- [103] Transaction Processing Performance Council, www.tpc.org, 2004.
- [104] J. Triplett, "Measuring Technological Change with Characteristics-Space Techniques," *Technological Forecasting and Social Change*, vol. 27, pp. 283-307, 1985.
- [105] M. D. Troutt, A. Rai, S. K. Tadisina, A. Zhang, "A New Efficiency Methodology for IT Investment Analysis Studies," *Measuring Information Technology*

Investment Payoff: Contemporary Approaches, Hershey, PA: IDEA Group, 1998.

- [106] Warwick-DEA, 1996.
- [107] Y.-H. B. Wong, J. E. Beasley, "Restricting weight flexibility in data envelopment analysis," *Journal of the Operational Research Society*, vol. 41, pp. 829-35, 1990.
- [108] J. Worlton, Some patterns of technological change in high-performance computers, 1988.
- [109] E. J. Wright, Comparing projectors using data envelopment analysis, 1999.
- [110] PriceWatch, www.pricewatch.com, 2004.
- [111] M. Xue, P. T. Harker, "Ranking DMUs with infeasible super-efficiency DEA models," *Management Science*, vol. 48, pp. 705-710, 2002.
- [112] R. Yin, "DEA: A New Methodology for Evaluating the Performance of Forest Products Producers," *Forest Products Journal*, vol. 48, pp. 29-34, 1997.
- [113] J. Zhu, "Super-efficiency and DEA sensitivity analysis," *European Journal of Operational Research*, vol. 129, pp. 443-455, 2001.
- [114] J. Zhu, DEAFrontier, 2003.
- [115] J. Zhu, "Data envelopment analysis with preference structure," *Journal of the Operational Research Society*, vol. 47, pp. 136-150, 1996.

Appendix A - TFDEA Crib Sheet

	Represents the output oriented rate of technological change at time t .
γ_k^t	Example: $\gamma_2 = 1.05$ Indicates that the SOA output should increase by 5% per time period with a given amount of input, based on $t=2$.
$\gamma_k^{\Delta t}$	Estimate of technological change based on product k 's relative to the SOA frontier at time Δt after k 's release.
	Represents the input oriented rate of technological change at time t .
β_k^t	Example: $\beta_2 = 0.95$ Indicates that the SOA input should decrease by 5% per time period with a given amount of output, based on the results at $t=2$.
$\beta_k^{\Delta t}$	Estimate of technological change based on product k 's relative to the SOA frontier at time Δt after k 's release.
ϕ_k^t	Output oriented technological index of product k at time of release. Output-orientation indicates that a technology aims to provide more functionality with the same structure. Example: $\phi_{0,k} = 1.05$ Indicates that product k should provide 5% more of each output to be state of the art.
$\theta_{0,k}$	Input-oriented technological index of product k at time of release. Input-orientation indicates that a technology aims to similar functionality with less structure. Example: $\theta_{0,k} = 0.95$ Indicates that product k at time of release should use 5% less of each input to be state of the art.
$\phi_{\Delta t}$	Output-oriented technological index of product k at Δt time units after its release.
$\theta_{\Delta t}$	Input-oriented technological index of product k at Δt time units after its release.

Measurement of State of The Art

$$\begin{aligned}
 & \min \theta_k \\
 & \text{s.t.} \quad \sum_{j=1}^n x_{i,k} \lambda_j \leq \theta_k x_{i,k}, \quad \forall i \in \{1 \dots m\} \\
 & \quad \quad \sum_{j=1}^n y_{r,j} \lambda_j \geq y_{r,k}, \quad \forall r \in \{1 \dots s\} \\
 & \quad \quad \lambda_j \geq 0, \\
 & \quad \quad \theta_k \geq 0.
 \end{aligned}$$

*Formula 59 - Input Oriented State of The Art
for Product k*

$$\begin{aligned}
 & \max \phi_k \\
 & \text{s.t.} \quad \sum_{j=1}^n x_{i,j} \lambda_j \leq x_{i,k}, \quad \forall i \in \{1, \dots, m\} \\
 & \quad \quad \sum_{j=1}^n y_{r,j} \lambda_j \geq \phi_k y_{r,k}, \quad \forall r \in \{1, \dots, s\} \\
 & \quad \quad \lambda \geq 0.
 \end{aligned}$$

*Formula 60 - Output-Oriented State of the Art for
Product k*

Technological Progress

$$\gamma_k^t = (\phi_k^t)^{\frac{1}{\Delta t}} \quad \theta_k^t = (\beta_k)^{(\Delta t)} \theta_k^{t_k}$$

Effective Time

$$t' = \frac{\sum_{j=1}^n (t_j - t_k) \lambda_{k,j}}{\sum_{j=1}^n \lambda_{k,j}}$$

Appendix B - TPC Dataset

<i>DMU</i>	<i>Platform</i>	<i>tpmC</i>	<i>Cost</i>	<i>Time</i>
1	IBM eServer iSeries 400 Model 840-2420	152346.25	6782752	09/15/00
2	ProLiant ML570-700 3P	20207.2	113798	09/26/00
3	IBM RS/6000 Enterprise Server M80 c/s	66750.27	2523482	09/30/00
4	IBM e(logo) server xSeries 250 c/s	32377.17	245460	10/25/00
5	hp server rp 7400	60366.82	1065085	12/01/00
6	PowerEdge 4400	16262.9	233532	12/01/00
7	IBM e(logo) xSeries 350 c/s	34264.9	276075	12/11/00
8	IBM eServer iSeries 400 Model 840-2420-001	163775.8	8448137	12/15/00
9	Compaq AlphaServer GS320 Model 6/731	155179.25	8205964	02/02/01
10	HP NetServer LXr 8500	43046.55	435038	03/01/01
11	IBM eServer pSeries 680 Model 7017-S85	220807.27	6469929	04/13/01
12	Compaq AlphaServer ES40.Model 6/833	37274	627144	05/01/01
13	Bull Escala EPC2450 c/s	220807.27	7657157	05/28/01
14	Bull Escala Epc 810 c/s	66750.27	2508189	05/28/01
15	IBM eServer pSeries 660	57346.93	1632624	06/19/01
16	PRIMEPOWER 2000 c/s w /32 Front Ends	222772.33	9671742	06/30/01
17	Bull Escala PL600R C/S	57346.93	1638835	07/03/01
18	Compaq AlphaServer GS320	230533	10286029	07/30/01
19	HP Netserver LH 6000 Client/Server	37596.34	333572	08/23/01
20	Sun Enterprise 4500	67102.53	1734522	09/14/01
21	hp server rp8400	140239.97	2015289	09/18/01
22	IBM eServer pSeries 660 Model 6M1	105025.02	2462401	09/21/01
23	ProLiant ML530-X1000-1P	9347.24	44582	09/25/01
24	Bull Escala PL800R	105025.02	2668861	09/26/01
25	ProLiant ML530-x1000-2P	17335.75	169758	09/26/01
26	Express5800/180Rb-7	52671.3	682724	09/30/01
27	IBM e(logo) xSeries 350 c/s	20422.01	110015	10/01/01
28	ProLiant DL580 6/900	39158.09	310945	10/15/01
29	IBM eServer xSeries 220 c/s	9112.91	43370	10/16/01
30	PowerEdge 2500/1.13/1P	11320.02	53203	10/31/01
31	IBM e(logo) xSeries 250 c/s	15533.72	72487	11/05/01
32	ProLiant ML570 6/900-4P	37100.52	216778	11/12/01
33	PowerEdge 2500/1.13/1P	11314.11	49484	12/14/01
34	PRIMEPOWER 2000 c/s w 66 Front-Ends	455818.2	12025524	02/28/02
35	Unisys e-@ction Enterprise Server ES7000	141138.44	3363483	03/10/02
36	Unisys e-@ction Enterprise Server ES7000	165218.71	3524109	03/10/02
37	PowerEdge 2500/1.26/1P	11537.02	42451	03/12/02
38	ProLiant DL760 8P/900	69169.61	651957	03/30/02
39	ProLiant ML370 T02/1.26-2P	17078.88	67996	03/30/02
40	AlphaServer ES45 MoDel 68/1000	50117	763829	05/09/02

DMU	Platform	tpmC	Cost	Time
41	HP 9000 Superdome Enterprise Server	389434.4	6388889	05/15/02
42	ProLaint ML530G2T 2P	34473.15	215905	05/28/02
43	IBM e(logo) xSeries 220 c/s	12009.46	42287	06/03/02
44	PowerEdge 4600/2.2/1P	12579.04	41583	06/26/02
45	ProLiant DL580-G2-4P	48911.83	320544	07/02/02
46	HP ProLiant ML530G2T-1P	17659.53	67574	07/25/02
47	Express5800/140Rb-4	48150.72	311093	07/31/02
48	Express5800/140Hc	48150.72	310721	07/31/02
49	IBM eServer xSeries 235/2.4GHz/1P	17559.31	52526	08/16/02
50	HP 9000 Model Superdome Enterprise Server	423414.41	6621072	08/26/02
51	IBM e(logo) xSeries 360 c/s	45230.03	204271	08/30/02
52	Fujitsu PRIMEPOWER 850	112286.46	1508712	08/31/02
53	IBM e(logo) xSeries 360 c/s	23027.66	101450	09/10/02
54	PowerEdge 2650/2.4/1P	16756.52	46502	09/12/02
55	AlphaServer ES45 Model 68/1250	56375	531965	09/27/02
56	Bull Escala PL3200R	403255.46	7245205	11/22/02
57	HP ProLiant DL580-G2/2GHz 4P	77905.18	413764	12/31/02
58	PowerEdge 6600/4/1.6GHz	51069.87	288627	12/31/02
59	HP ProLiant DL580-G2-32GB	61564.5	376806	12/31/02
60	HP ProLiant ML350G3T-1P	10089.76	36214	12/31/02
61	HP ProLiant ML570-G2 4P	68739.22	341990	12/31/02
62	HP ProLiant DL380-G3	18051.65	60965	02/01/03
63	IBM eServer xSeries 225/2.4Ghz/1P	18077.98	50509	02/12/03
64	rx5670	87741	441022	02/12/03
65	PowerEdge 6650/4/2.0Ghz/32GB	71586.49	364887	03/31/03
66	PowerEdge 6600/4/2.0Ghz/32GB	71313.19	365317	03/31/03
67	HP ProLiant ML530G2T-2P	38386.24	236876	03/31/03
68	HP ProLiant ML 370G3-2P	26725.34	99211	03/31/03
69	IBM eServer xSeries 360/1.9Ghz/4P	50666.11	216838	04/30/03
70	IBM eServer xSeries 360/2.0GHz/4p	52587.46	226504	04/30/03
71	NEC Express5800/140Rc-4	70653.01	347076	04/30/03
72	Unisys ES7000 Orion 230 Enterprise Server	203518.03	2681773	05/02/03
73	Unisys ES7000 Orion 230 Enterprise Server	234325.1	2715310	05/02/03
74	HP ProLiant ML350G3-2P	39006.54	187141	05/12/03
75	HP ProLiant ML350G3-1P	19526.27	43826	05/12/03
76	HP ProLiant DL380G3-2P	43230.66	160353	05/27/03
77	HP ProLiant DL380G3-1P	18818.46	43502	05/27/03
78	HP ProLiant ML370G3-1P	19140.72	44548	05/29/03
79	HP ProLiant ML 370G3-2P	44942.92	219802	05/29/03
80	Unisys ES7000 Aries 520 Enterprise Server	118381.38	657533	05/30/03
81	HP ProLiant DL760-G2 8P	115025.75	884216	05/30/03
82	Unisys ES7000 Orion 540 Enterprise Server 16F	161542.04	986622	05/30/03

DMU	Platform	tpmC	Cost	Time
84	IBM eServer xSeries 235 1P c/s	18936.05	46539	07/10/03
85	HP ProLiant ML370G3-1M-2P	52468.48	200301	07/15/03
86	HP ProLiant ML370G3-1M-1P	19718.01	45478	07/15/03
87	Unisys ES7000 Orion 540 Enterprise Server	252920.49	1824732	07/22/03
88	hp server rx5670	121065.13	543023	08/01/03
89	PRIMERGY TX300	53691.33	205056	09/01/03
90	eServer xSeries 440 c/s	151744.13	1674017	09/25/03
91	HP ProLiant DL580G2/2.8GHz-4P	84712.94	324423	09/26/03
92	IBM eServer xSeries 440 c/s 8p	119115.16	781556	10/04/03
93	ProLiant ML370-G3-1M 2P	54096.56	203461	10/13/03
94	RackSaver RS-2164/op-r	20477.37	42266	10/21/03
95	QuatreX-64 Server 4P	82226.46	223664	10/21/03
96	PowerEdge 6600/4/2.0GHz/32GB	78116.87	378356	10/21/03
97	HP ProLiant DL380-G3-1M/3.20GHz-2p	51226.96	163924	10/22/03
98	HP Integrity Superdome	707102	5064831	10/23/03
99	HP Integrity Superdome	786646	5105486	10/23/03
100	IBM eServer pSeries 690 Turbo 7040-681	763898.39	6301741	11/08/03
101	IBM eServer pSeries 690 Turbo 7040-681	680613.12	7574961	11/08/03
102	HP ProLiant DL380-G3-1M/3.20GHz-1P	19814.35	44296	11/24/03
103	NEC Express5800/1320Xd c/s w/Express5800/12	577530.77	6238492	12/01/03
104	IBM eServer xSeries 235/3.2GHz/1P	31910.24	68477	12/03/03
105	HP ProLiant DL380-G3-1M/3.20GHz/1P-12GB	33873.83	81177	12/17/03
106	HP ProLiant ML350T03 X3.06/533	17192.4	33692	12/17/03
107	PowerEdge 6600/4/2.8GHz/32GB	84595.22	302148	12/30/03
108	IBM eServer xSeries 445 8P c/s	139153.98	705115	12/31/03
109	IBM eServer xSeries 445-16P c/s	190510.02	1599308	12/31/03
110	IBM eServer xSeries 445 4P c/s	90271.76	357969	12/31/03
111	PowerEdge2650	20108.79	41250	01/14/04
112	Unisys ES7000 Aries 420 Enterprise Server	309036.53	1384981	01/30/04
113	HP 9000 Superdome Enterprise Server	541673.76	6315689	01/30/04
114	PowerEdge 2650/1/3.2GHz/2M	22052	40775	02/18/04
115	PowerEdge 2650/1/3.2GHz/2M	32185.33	61788	02/23/04
116	IBM eServer xSeries 365 4P c/s	89616.32	333788	02/27/04
117	IBM eServer pSeries 690 Turbo 7040-681	768839.4	6574014	02/29/04
118	HP ProLiant DL580G2/3.0GHz-4P	95163	278114	03/02/04
119	HP Integrity rx5670 Linux	136110.98	556853	03/05/04
120	HP ProLiant ML350-G3-1M/2.80GHz-1P/8GB	28711	61399	03/17/04
121	HP ProLiant ML370-G3-2M/3.20GHz-1P	35030	65528	03/17/04
122	HP ProLiant ML370-G3-2M/3.20GHz-2P	60364	211519	03/17/04
123	IBM eServer xSeries 365 4P c/s	102667.42	361742	03/31/04

DMU	Platform	tpmC	Cost	Time
124	HP Integrity Superdome	1008144.49	8397262	04/14/04
125	HP rx8620	301225	1264031	04/15/04
126	Unisys ES7000 Aries 420 Enterprise Server	291410.61	1448233	04/20/04
127	HP Integrity rx5670 Cluster 64P	1184893.38	6541770	04/30/04
128	PRIMEPOWER 2500	595702.31	7399502	04/30/04
129	IBM eServer pSeries 690 Model 7040-681	1025486.07	5571349	08/16/04
130	IBM eServer xSeries 445 8P c/s	156105.72	672287	08/31/04

Appendix C - Moore's Law Dataset

OBS #	Company	Processor	Ship Date	SPEC95 Int	SPEC95 Fp	Tech (um)	Power (W)	Size (mm ²)	Tran. (10 ⁶)
16	PA-RISC	PA7100	2	3.2	4	0.8	23	196	0.8
13	Alpha	Alpha 21064	2	3	3.6	0.75	30	299	9.3
1	SPARC	Super	2	1.5	1.7	0.6	14.2	315	5.2
39	Alpha	Alpha 21164	4	8.5	12.7	0.5	50	209	9.3
18	PA-RISC	PA7150	4	5.2	4.6	0.8	30	259	9
20	PA-RISC	PA7100LC	4	4.6	4.7	0.8	10	345	3.9
27	Alpha	Alpha 21064a	4	5.2	6.3	0.5	33	138	3.5
6	Intel	P54C	4	3.3	2.8	0.6	5	148	3.1
5	Intel	P54VRT (Mot	4	2.9	2.5	0.6	6.5	150	3.1
3	Intel	P54VRT (Mot	4	2.4	2.1	0.6	5.2	150	3.1
28	Intel	P6 -2000MHz	5	8.2	6.8	0.35	35	195	5.5
25	Power PC	PC 620	5	6	6	0.5	30	67	6.35
23	Intel	P6 -150MHz	5	6.1	5.4	0.6	29.2	195	5.5
8	Intel	P54CQS	5	3.8	3	0.35	10	90	3.1
37	SPARC	Ultra I+	5	7.8	11.4	0.42	30	156	5.4
35	SPARC	Ultra I	5	6.6	9.4	0.47	30	149	5.4
14	Power PC	PC 604 -133M	5	4.7	3.8	0.5	5.6	311	7
11	SPARC	Super 2	5	3.5	3.5	0.6	15	265	5.2
2	SPARC	Micro 2	5	1.6	2	0.4	9	299	3.1
26	Intel	P6 -166MHz	6	7.3	6.2	0.35	29.4	131	7.5
43	SPARC	Ultra II	6	10.4	15	0.35	25	132	3
19	Power PC	PC 603e -24C	6	6.3	4.6	0.35	5.5	197	3.6
10	Intel	P54CS-166M	6	4.8	3.3	0.35	10	90	3.1
24	MIPS	R5000	6	5.5	5.5	0.35	10	596	3.43
7	SPARC	Turbo	6	3.5	3	0.35	9	162	8.8
48	MIPS	R10000	6	11.4	19.1	0.35	30	196	0.85
9	Intel	P54CS-150M	6	4.3	3.3	0.35	10	90	3.1
47	Alpha	Alpha 21164a	6	12.3	17.2	0.35	30.5	350	100
36	Power PC	x704	7	12	10	0.5	85	83	6.5
49	Alpha	Alpha 21164 I	7	17.3	19.9	0.35	40.5	302	15.2
44	MIPS	RM7000	7	13	15	0.25	13	298	6.8
12	AMD	K6 - 200MHz	7	6.2	3.5	0.35	20	184	22
21	Intel	P55C (MMX)	7	6.4	4.7	0.35	15.7	141	4.5
22	Intel	P55C (MMX)	7	7.1	5.2	0.35	17	141	4.5
15	AMD	K6 -233MHz	7	6.8	3.8	0.35	28.3	184	22
4	CYRIX	6x86 MX (MII	7	5.1	2.2	0.35	10	162	8.8
17	Intel	P55VRT (MM	7	5.6	4.3	0.35	7.8	141	4.5
32	Intel	Pentium II (KI	7	11.9	8.6	0.35	43	131	7.5
31	Power PC	PC 604e -30C	7	12.9	8.5	0.25	12	150	2.7
55	Alpha	Alpha 21264	8	44	66	0.35	72	78	1.4
46	SPARC	Ultra 2i	8	14.2	16.9	0.35	30	197	6
29	Power PC	G3 740	8	12.2	7.1	0.25	4.2	234	1.68
34	Intel	Celeron (Desi	8	11.9	8.6	0.25	17.5	131	9.5
33	Intel	Pentium II (De	8	11.9	8.6	0.25	26	131	7.5
41	Intel	Pentium II (Xe	8	16.5	13.7	0.25	23.3	131	7.5
42	Intel	Pentium II (Xe	8	18.9	14.7	0.25	23.3	131	7.5
52	AMD	Athlon - K7 - !	9	23.6	20.6	0.25	46	164	1.2
53	AMD	Athlon - K7 -7	9	31.7	24	0.25	50	198	2.4
54	Intel	Pentium III (C	9	35.6	30.4	0.18	34	106	28
38	Intel	Celeron (Men	9	15.1	11.8	0.25	23.4	225	19
45	Intel	Pentium III (K	9	24	15.9	0.25	34	106	9.5
40	Intel	Celeron (Men	9	17.9	12.9	0.25	23.4	225	19
51	Power PC	G4 7400	9	21.4	20.4	0.15	8	164	2.8

Appendix D - TFDEA Software

The TFDEA php scripts were created to provide a means to solve DEA through a web page. They were written to be used with RedHat Linux 9.0 as well as Fedora Core 1, 2, and 3.

Software Required

Prior to using the software developed for this research the following packages are required for use.

1. mysql – mysql access to the data repository
2. mysql-server – the data repository
3. php – scripting language used to perform TFDEA
4. php-mysql – the interface between php and mysql
5. glpk – gnu linear programming kit used to solve the linear equations
6. php-glpk – gnu linear programming kit used as the interface between glpk and php

Once the above are installed, the /etc/php.ini file must be modified to increase the maximum time which a script can run this is denoted by the variable:

`max_execution_time` which should be set to 300 (five minutes) for larger problem sets.

Software Components

The TFDEA software was created to simplify creation of a web portal to solve TFDEA and DEA problems it is composed of a number of scripts, which are summarized below.

1. AForecast.php – This program does the forecast for DEA. This is executed through the command line `php AForecast.php`.
2. DEAFuctions.php – This library works with GLPK to perform DEA.
3. MathFunctions.php – This library performs various math functions required by TFDEA.
4. DataFunctions.php – This library is responsible for obtaining data from the data repositories for analysis.
5. LogFunctions.php – This library is responsible for logging the output of the programs.
6. Config.php – This script is used to provide database configuration and output configuration parameters.

AForecast.php

```
<?php
//
// Project:  TFDEA Script
//
// Author:   Lane Inman
// $Id: AForecast.php,v 1.54 2004/11/09 00:08:25 oli Exp $
//

//
// Here if there is an initialization to be done we will reexamine.
//
//if ( ! $webinit ) {
//} else {
//    session_start();
```

```

//}

//
// Include functions here
//
include "LogFunctions.php" ;
// include "ChartFunctions.php" ;
include "DEAFunctions.php" ;
include "MathFunctions.php" ;
include "DataFunctions.php" ;
include "Config.php" ;
include "LogLevels.php" ;

//
// Here we give an opportunity for an include file - I did this to remove the
// non important changes from the structure (IE Table, etc)
//
//if ( $file != " " ) {
include "Projectors.php" ;
// include "Fighters.php" ;
// include "TPC.php";
//include "DT.php";
// include "PenDrive.php";
//}

//
// Define the name of the script for the log functions
//
$functionName = "AForecast.php" ;

$DMUDATA = $newDMUs = array();

//
// Get DMU Information from the database
//
logTrace( $functionName, 8, "Connecting to database." );
if ( !$connection = @mysql_connect($serverName, $username, $password)) {
    $variables = "stage=setup_db_connection"
        . "&fail=yes"
        . "&serverName=$serverName"
        . "&username=$username";
    header("location: ?$variables");
} else {
    mysql_select_db($database);
    $DMUDATA=getData2($connection, $tableName, $iTimeUnits,
        $Inputs, $Outputs, $InputsAs, $OutputsAs, $timeColumn,
        $whereStatement );
}

//
// Init the variables because if we dont we are fragged
//
$Time          = $Iter          = 0;
$soaDMUs       = $periodNewSOA = $newSquare = array();
$newSOAs       = array();
$DMUs          = array_keys($DMUDATA);

//
// Solve the problem based on the model chosen
//
foreach ( array_keys($DMUDATA) as $RowName ) {
    $Row          = $DMUDATA[$RowName];
    $Observation = array();
    $Iter         += 1;
}

//

```

```

// Set the time to the current time if there is no other time
//
if ( $Time == 0 ) { $Time=$Row["Time"]; }

//
// if the time is different, then we can calculate the state of the
// art.
//
if ( $Row["Time"] != $Time ) {
//
// The number of DMUs at this time are set
//
$newDMUCount[$Time]          = count($newDMUs);

//
// amend to the current soaDMUs the current DMUs
//
$currentDMUs = array_merge($soaDMUs, $newDMUs);

$newScores[$Time][INPUT_ORIENTATION] = 0;
$newSquare[$Time][INPUT_ORIENTATION] = 0;

$newScores[$Time][OUTPUT_ORIENTATION] = 0;
$newSquare[$Time][OUTPUT_ORIENTATION] = 0;

//
// Analyze every DMU
//
foreach ( $currentDMUs as $DMU ) {
    $OOResult = CCR( $DMU          , $bSuper          ,
                    $Inputs       , $Outputs        ,
                    $currentDMUs , $iRTS           ,
                    $iDebug       , $DMU_Data        ) ;
    $IOResult = IO( $DMU          , $bSuper          ,
                   $Inputs       , $Outputs        ,
                   $currentDMUs , $iRTS           ,
                   $iDebug       , $DMU_Data        ) ;

//
// This is for the goodness of fit data...
//
if ( $bGoodFit && in_array( $DMU, $newDMUs ) ) {
    $newScores[$Time][INPUT_ORIENTATION] += $IOResult["Theta"];
    $newSquare[$Time][INPUT_ORIENTATION] +=pow($IOResult["Theta"],
        2);
    $newScores[$Time][OUTPUT_ORIENTATION] += $OOResult["Theta"];
    $newSquare[$Time][OUTPUT_ORIENTATION] +=pow($OOResult["Theta"],
        2);
}

//
// This is here because of the integer vs. real issue...
//
if ( ( $iOrientation == OUTPUT_ORIENTATION &&
    $OOResult["Theta"] <= ( 1 + CLOSE_ENOUGH ) ) ||
    ( $iOrientation == INPUT_ORIENTATION &&
    $IOResult["Theta"] >= ( 1 - CLOSE_ENOUGH ) ) ) {
    if ( ! in_array($DMU, $soaDMUs) ) {
        $soaDMUs      = array_merge($soaDMUs, array($DMU));
        $bNewSOA      = 1 ;
    }
    $newSOAs=array_merge($newSOAs, array($DMU));
} elseif ( $bProxy && in_array($DMU, $newDMUs) ) {
    foreach ( $newDMUs as $proxyDMU ) {
        logTrace($functionName, 9,
            "Determining proxy at time $Time for "
            . $proxyDMU);
        $DMUData[$newDMU] = buildProxyDMU ( $proxyDMU

```

```

$DMUData      ,
$iOrientation  ,
$Inputs       ,
$iOutputs     ,
$ResultArray  );
    }
}

if ( $DMU_Data[$DMU]["Time"] != $Time ) {
    logTrace($functionName, 21, "Time: $Time");
    $OOProgress[$DMU]=AnalyzeProgress( $DMU      , $Time      ,
                                       OUTPUT_ORIENTATION
                                       $DMU_Data  , $OOResult  ,
                                       $soaDMUs  , $ROctype  ,
                                       $iRTS     );

    $IOProgress[$DMU]=AnalyzeProgress( $DMU      , $Time      ,
                                       INPUT_ORIENTATION
                                       $DMU_Data  , $IOResult  ,
                                       $soaDMUs  , $ROctype  ,
                                       $iRTS     );
}
$OOSolution[$DMU]=$OOResult;
$IOSolution[$DMU]=$IOResult;
}
//
// If we have new SOA then we will record the information, otherwise
// nothing has changed from the last point in time.
//
if ( $bDisplayPeriod ) {
    $stamp = getStamp( $Time, $connection ) ;

    reportMessage("=====");
    reportMessage("Time Period $stamp ( $Time ) Results" );
    reportMessage("=====");
    if ( $iOrientation == OUTPUT_ORIENTATION ) {
        ShowDEAResults( $displayFormat, $iOrientation,
                        $currentDMUs , $OOSolution );
    } elseif ( $iOrientation == INPUT_ORIENTATION ) {
        ShowDEAResults( $displayFormat, $iOrientation,
                        $currentDMUs, $IOSolution );
    }
}
if ( $bNewSOA ) {
    //
    // Find the progress statistics here
    //
    logTrace($functionName, 1, "Calling FindStats Time " .
            "is $Time");
    $OsumStats[$Time] = FindStats( $OOProgress,
                                   OUTPUT_ORIENTATION );
    $IsumStats[$Time] = FindStats( $IOProgress,
                                   INPUT_ORIENTATION );
    $periodNewSOA[$Time] = $newSOAs ;
    $soaPeriods[$Time] = $Time ;
}

$bNewSOA = 0 ;
$newSOAs = $newDMUs = array();
$Time=$Row["Time"];
}

foreach ( array_merge($Inputs,$Outputs) as $Factor ) {
    $Observation[$Factor] = $Row[$Factor];
}

$OOSolution = $IOSolution = $OOProgress = $IOProgress = array();
$Observation["Time"] = $Row["Time"];

```



```

        $DMU_Data[$Row["DMU"]]= $Observation;
        $newDMUs=array_merge($newDMUs,$Row["DMU"]);
    }

    $OOResult=$IOResult=$OOProgress=$IOProgress=array();
    //
    // The number of DMUs at this time are set
    //
    $newDMUCount[$Time]                = count($newDMUs);

    //
    // amend to the current soaDMUs the current DMUs
    //
    $currentDMUs=array_merge($soaDMUs, $newDMUs);

    $newScores[$Time][INPUT_ORIENTATION] = 0;
    $newSquare[$Time][INPUT_ORIENTATION] = 0;

    $newScores[$Time][OUTPUT_ORIENTATION] = 0;
    $newSquare[$Time][OUTPUT_ORIENTATION] = 0;

    foreach ( $currentDMUs as $DMU ) {
        $OOResult = CCR( $DMU, $bSuper, $Inputs, $Outputs, $currentDMUs,
            $iRTS, $iDebug, $DMU_Data );
        $IOResult = IO( $DMU, $bSuper, $Inputs, $Outputs, $currentDMUs,
            $iRTS, $iDebug, $DMU_Data );

        //
        // This is for the goodness of fit data...
        //
        if ( $bGoodFit && in_array( $DMU, $newDMUs ) ) {
            $newScores[$Time][INPUT_ORIENTATION] += $IOResult["Theta"];
            $newSquare[$Time][INPUT_ORIENTATION] += pow($IOResult["Theta"],
                2);

            $newScores[$Time][OUTPUT_ORIENTATION] += $OOResult["Theta"];
            $newSquare[$Time][OUTPUT_ORIENTATION] += pow($OOResult["Theta"],
                2);
        }

        //
        // This is here because of the integer vs. real issue....
        //
        if ( ( $iOrientation == OUTPUT_ORIENTATION &&
            $OOResult["Theta"] <= ( 1 + CLOSE_ENOUGH ) ) ||
            ( $iOrientation == INPUT_ORIENTATION &&
            $IOResult["Theta"] >= ( 1 - CLOSE_ENOUGH ) ) ) {
            if ( ! in_array($DMU, $soaDMUs) ) {
                $soaDMUs      = array_merge($soaDMUs, array($DMU));
                $bNewSOA      = 1 ;
            }
            $newSOAs=array_merge($newSOAs, array($DMU));
        }
        if ( $DMU_Data[$DMU]["Time"] != $Time ) {
            $OOProgress[$DMU]=AnalyzeProgress( $DMU
                , $Time
                , OUTPUT_ORIENTATION, $DMU_Data,
                $OOResult
                , $soaDMUs
                , $ROctype
                , $iRTS
                );

            $IOProgress[$DMU]=AnalyzeProgress( $DMU, $Time, INPUT_ORIENTATION,
                $DMU_Data, $IOResult, $soaDMUs,
                $ROctype, $iRTS);
        }
        $OOSolution[$DMU]=$OOResult;
        $IOSolution[$DMU]=$IOResult;
    }
    //
    // If we have new SOA then we will record the information, otherwise
    // nothing has changed from the last point in time.
    //

```

```

if ( $bDisplayPeriod ) {
    $stamp = getStamp( $Time, $connection );
    reportMessage("=====");
    reportMessage("Time Period $stamp ( $Time ) Results");
    reportMessage("=====");
    if ( $iOrientation == OUTPUT_ORIENTATION ) {
        ShowDEAResults( $displayFormat, $iOrientation,
            $currentDMUs, $OOSolution );
    } elseif ( $iOrientation == INPUT_ORIENTATION ) {
        ShowDEAResults( $displayFormat, $iOrientation,
            $currentDMUs, $IOSolution );
    }
}

if ( $bNewSOA ) {
    //
    // Find the progress statistics here
    //
    logTrace($functionName, 40, "Time is $Time");
    $OsumStats[$Time] = FindStats( $OOProgress, OUTPUT_ORIENTATION );
    $IsumStats[$Time] = FindStats( $IOProgress, INPUT_ORIENTATION );
    $periodNewSOA[$Time] = $newSOAs ;
    $soaPeriods[$Time] = $Time ;
}

//
// Rate Of Change Results
//
logTrace($functionName, 1000300, "Checking whether goodness of fit...");
if ( $bGoodFit ) {
    $oFunctionName = $functionName ;
    $functionName = "calculateGoodness";
    logTrace($functionName, 1000300, " Checking goodness of fit...");
    tableHeader( "Average Efficiency", "Time Period", array("Input","Output"));

    $rowCount = 0;
    $sumScore[INPUT_ORIENTATION] = $sumScore[OUTPUT_ORIENTATION] = 0;
    $sumSquares[INPUT_ORIENTATION] = $sumSquares[OUTPUT_ORIENTATION] = 0;
    $total[INPUT_ORIENTATION] = $total[OUTPUT_ORIENTATION] = 0;
    $count[INPUT_ORIENTATION] = $count[OUTPUT_ORIENTATION] = 0;

    foreach ( array_keys($newScores) as $TimePeriod ) {
        $avgScore[INPUT_ORIENTATION] =
            $newScores[$TimePeriod][INPUT_ORIENTATION] /
            $newDMUCount[$TimePeriod];
        $avgScore[OUTPUT_ORIENTATION] =
            $newScores[$TimePeriod][OUTPUT_ORIENTATION] /
            $newDMUCount[$TimePeriod];
        $sumScore[INPUT_ORIENTATION] +=
            $newScores[$TimePeriod][INPUT_ORIENTATION] ;
        $sumScore[OUTPUT_ORIENTATION] +=
            $newScores[$TimePeriod][OUTPUT_ORIENTATION] ;
        logTrace( $functionName, 1000300, "TimePeriod [" . $TimePeriod
            . "] Input [" . $newSquare[$TimePeriod][INPUT_ORIENTATION]
            . "]" );
        logTrace( $functionName, 1000301, "Time Period [" . $TimePeriod
            . "] Output [" . $newSquare[$TimePeriod][OUTPUT_ORIENTATION]
            . "]" );
        $sumSquares[INPUT_ORIENTATION] +=
            $newSquare[$TimePeriod][INPUT_ORIENTATION] ;
        $sumSquares[OUTPUT_ORIENTATION] +=
            $newSquare[$TimePeriod][OUTPUT_ORIENTATION] ;

        $displayTime=getStamp($TimePeriod, $connection );
        //
        // If we want verbose report, report it all
        //
        if ( ! $bSummary ) {

```

```

        tableRow($displayTime , $avgScore, $rowCount, 0);
    }
    if ( $avgScore[INPUT_ORIENTATION] > 0 ) {
        $total[INPUT_ORIENTATION] += $avgScore[INPUT_ORIENTATION];
        $count[INPUT_ORIENTATION] += $newDMUCount[$TimePeriod];
    }
    if ( $avgScore[OUTPUT_ORIENTATION] > 0 ) {
        $total[OUTPUT_ORIENTATION] += $avgScore[OUTPUT_ORIENTATION] ;
        $count[OUTPUT_ORIENTATION] += $newDMUCount[$TimePeriod];
    }
    $rowCount++;
}
tableRow("Sum Averages ", $total, $rowCount, 0 );
$rowCount++;
tableRow("Sum Scores   ", $sumScore, $rowCount, 0 );
$rowCount++;
tableRow("Sum Squares  ", $sumSquares, $rowCount, 0);
$rowCount++;
tableRow("Total Count  ", $count, $rowCount, 0 );
$rowCount++;
$total[OUTPUT_ORIENTATION] = $sumScore[OUTPUT_ORIENTATION] /
    $count[OUTPUT_ORIENTATION];
$total[INPUT_ORIENTATION] = $sumScore[INPUT_ORIENTATION] /
    $count[INPUT_ORIENTATION];
tableRow("Avg Summary: ", $total, $rowCount, 0);
tableFooter("");
$functionName = $oFunctionName ;
}

logTrace($functionName, 1000200, "Checking to display ROC...");
if ( $bDisplayROC ) {
    logTrace($functionName, 1000201, "Displaying ROC...");
    tableHeader("Rates of Change", "Time", array("Gamma", "Conf", "Beta", "Conf"));
    $rowCount = 0;
    foreach ( array_keys($OsumStats) as $Period ) {
        $timeToPrint = getStamp( $Period, $connection );
        tableRow( $timeToPrint, array($OsumStats[$Period][0] ,
            $OsumStats[$Period][2] ,
            $IsumStats[$Period][0] ,
            $IsumStats[$Period][2]),
            $rowCount
            ,
            0
        ) ;

        $rowCount++;
    }
    tableFooter("");
}

logTrace($functionName, 1000201, "Checking Graphing ROC...");
if ( $bGraphROC ) {
    logTrace($functionName, 1000201, "Graphing ROC...");
    ShowROC ( $OsumStats, $iOrientation, "Quarterly", 0, $connection );
}

//
// Here we are setting up the end of the forecast ( we will not validate
// any forecasts beyond this point because they are after the present )
//
logTrace($functionName, 1000201, "Setting Predict Base...");
if ( $PredictBase != "" ) {
    $stopTime = getPeriod($PredictBase, $connection);
} else {
    $stopTime = 10000000000 ;
}

logTrace($functionName, 1000201, "Checking Validate...");
if ( $bValidate ) {
    $functionName = "Validate";
}

```

```

if ( ! $bSummary ) {
    tableHeader( "Forecast Results", "Time", array("In Zone"      ,
                                                    "Only Lower"     ,
                                                    "Avg Overage"    ,
                                                    "Only Upper"     ,
                                                    "Avg Underage"   ,
                                                    "Not Feasible"  ,
                                                    "Broken"        ));
}

$Underage = $Overage = $rowCount = 0;
$total    = $Good    = $UpperGood = $LowerGood = $Bad    = $Hosed = 0;

$tTotal   = $tGood   = $tUpperGood = $tLowerGood = $tBad = $tHosed = 0;

$HosedString = $BList = "";
$LowerInfeasible = 0;

foreach ( $soaPeriods as $soaPeriod ) {
    foreach ( array_keys($periodNewSOA) as $evalPeriod ) {
        if ( $evalPeriod > $soaPeriod && $evalPeriod <= $stopTime ) {
            $soaDate = getStamp( $soaPeriod, $connection );
            foreach ( $periodNewSOA[$evalPeriod] as $newDMU ) {
                if ( ! in_array($newDMU, $periodNewSOA[$soaPeriod]) &&
                    $DMU_Data[$newDMU]["Time"] == $evalPeriod ) {
                    $evalDate = getStamp($evalPeriod, $connection);
                    logTrace($functionName, 4, "Calling EvaluateForecast");
                    $Forecast = EvaluateForecast($soaPeriod
                                                ,
                                                $iOrientation
                                                ,
                                                $iRTS
                                                ,
                                                $iValidate
                                                ,
                                                $ROctype
                                                ,
                                                $periodNewSOA[$soaPeriod],
                                                $newDMU
                                                ,
                                                $IsumStats[$soaPeriod],
                                                $OsumStats[$soaPeriod],
                                                $Inputs
                                                ,
                                                $Outputs
                                                ,
                                                $DMU_Data
                                                ,
                                                $bValidateTrace
                                                );
                    //if ( $bValidateTrace && $Forecast["Lower"] != -5 ) {
                    logTrace($functionName, 1, "soaPeriod ["
                        . $soaPeriod . "] soaDate ["
                        . $soaDate . "] evalPeriod ["
                        . $evalPeriod . "] newDMU ["
                        . $newDMU . "] Lower ["
                        . $Forecast["Lower"] . "] Upper ["
                        . $Forecast["Upper"] . "]" );
                    //}

                    if ( $Forecast["Lower"] == 1 &&
                        ( $Forecast["Upper"] == 0 ||
                          $Forecast["Upper"] == -20 ) ) {
                        $LGList .= $newDMU . ":" . $evalDate . " ";
                        $LowerGood++;
                        logTrace($functionName, 2, "-->OnlyLower");
                        $Overage += $Forecast["UpperScore"] ;
                    } elseif ( $Forecast["Upper"] == 1 &&
                              $Forecast["Lower"] == 0 ) {
                        //|| $Forecast["Lower"] == -20 ) {
                        $UGList .= $newDMU . ":" . $evalDate . " ";
                        $UpperGood++;
                        logTrace($functionName, 2, "-->OnlyUpper");
                        logTrace($functionName, 1, "    Upper good");
                        if ( $Forecast["Lower"] == -20 ) {
                            $LowerInfeasible .= $newDMU . ":" . $soaDate
                                . "@" . $evalDate . "("
                                . $soaPeriod . "@"

```



```

        $tGood      += $Good      ;
        $tUpperGood += $UpperGood;
        $tLowerGood += $LowerGood;
        $tBad       += $Bad       ;
        $tHosed     += $Hosed     ;
    }

    $Good          = $Bad          = $UpperGood = $LowerGood = 0 ;
    $Hosed         = $Total        = $Underage  = $Overage   = 0 ;

    $GList        = $UGList       = $LGList    = $HList     = "";
    $rowCount++;
}
if ( ! $bSummary ) { tableFooter(""); }

$tTotal = $tGood + $tLowerGood + $tUpperGood + $tBad + $tHosed ;

tableHeader("Forecasting Summary", "Action", array("Number", "Percentage"));
tableRow("Good Forecasts", array( $tGood,
                                $tGood/$tTotal * 100 ),
        0, 0 );
tableRow("Good Lower Forecasts", array( $tLowerGood,
                                       $tLowerGood/$tTotal * 100 ),
        0, 0 );
if ( $overageCount ) {
    tableRow("Overage Average", array( $overageSum/$overageCount,
                                       $overageCount ),
            0, 0 );
}
tableRow("Good Upper Forecasts", array( $tUpperGood,
                                       $tUpperGood/$tTotal * 100 ),
        0, 0 );
if ( $underageCount ) {
    tableRow("Underage Average", array( $underageSum/$underageCount,
                                       $underageCount ),
            0, 0 );
}
tableRow("Not Forecasted", array( $tBad,
                                 $tBad/$tTotal * 100 ),
        0, 0 );
tableRow("Total Hosed*", array( $tHosed,
                               $tHosed/$tTotal * 100 ),
        0, 0 );
tableRow("Total", array( $tTotal,
                        $tTotal/$tTotal * 100 ),
        0, 0 );

tableFooter("");

reportMessage("Not Predicted: " . $BList );
reportMessage("Lower Infeasible: " . $LowerInfeasible );
reportMessage("HOSED LIST: " . $HosedString );
$functionName="AForecast.php";
}

logTrace($functionName, 1000201, "Checking Futures...");
if ( $bCheckFutures ) {
    logTrace($functionName, 1000201, "Running Futures Check...");
    $reportStatus = array (FORECAST_ABOVE_LOWER => "Early Product",
                          FORECAST_BELOW_UPPER => "Late Product",
                          FORECAST_GOOD      => "Expected Release" ,
                          FORECAST_NOT_COVERED => "Outside of Forecast" ,
                          FORECAST_ERROR     => "Error with Forecast" );

    $assessment=array();
    foreach ( array_keys($soaPeriods) as $keyToCheck ) {
        //
        // If we are below the stop time continue
        //

```

```

if ( $soaPeriods[$keyToCheck] <= $stopTime ) {
    $lastRealSOA = $keyToCheck ;
} else {
    break ;;
}
$soaPeriod = $soaPeriods[$lastRealSOA];
}
foreach ( array_keys($DMU_Data) as $newDMU ) {
    $DMU_Name[$newDMU] = getField( 'DMU', $newDMU, $descriptorColumn,
        $tableName, $connection);
    if ( $DMU_Data[$newDMU]["Time"] > $soaPeriod ) {
        $evalDate = $DMU_Data[$newDMU]["Time"] ;
        $Forecast = EvaluateForecast($soaPeriod
            ,
            $iOrientation
            ,
            $iRTS
            ,
            $iValidate
            ,
            $ROctype
            ,
            $periodNewSOA[$soaPeriod],
            $newDMU
            ,
            $IsumStats[$soaPeriod]
            ,
            $OsumStats[$soaPeriod]
            ,
            $Inputs
            ,
            $Outputs
            ,
            $DMU_Data
            ,
            0
            ) ;

        $assessment[$newDMU]["LowerScore"]=$Forecast["LowerScore"];
        $assessment[$newDMU]["UpperScore"]=$Forecast["UpperScore"];

        if ( ( $Forecast["Lower"] == 1 || $Forecast["Lower"] == -20 ) &&
            ( $Forecast["Upper"] == 0 || $Forecast["Upper"] == -20 ) ) {
            $assessment[$newDMU]["conclusion"] = FORECAST_ABOVE_LOWER ;
        } elseif ( $Forecast["Upper"] == 1 &&
            $Forecast["Lower"] == 0 ) {
            $assessment[$newDMU]["conclusion"] = FORECAST_BELOW_UPPER ;
        } elseif ( ( $Forecast["Lower"] == 1 || $Forecast["Lower"] == -20 )
            && $Forecast["Upper"] == 1 ) {
            $assessment[$newDMU]["conclusion"] = FORECAST_GOOD ;
        } elseif ( ( $Forecast["Lower"] == -20 &&
            $Forecast["Upper"] == -20 ) ) { // ||
            $assessment[$newDMU]["conclusion"] = FORECAST_NOT_COVERED ;
        } elseif ( $Forecast["Lower"] == 0 && $Forecast["Upper"] == 0 ) {
            $assessment[$newDMU]["conclusion"] = FORECAST_ERROR ;
        }
        }

        $IROC=$IsumStats[$soaPeriod][0] ;
        $OROC=$OsumStats[$soaPeriod][0] ;

        if ( $iOrientation == OUTPUT_ORIENTATION ||
            $iValidate == VALIDATE_BOTH ) {
            $assessment[$newDMU]["estTimeO"] = estimateTime(
                $newDMU
                ,
                $periodNewSOA[$soaPeriod],
                $Inputs
                ,
                $Outputs
                ,
                $iRTS
                ,
                $DMU_Data
                ,
                $soaPeriod
                ,
                OUTPUT_ORIENTATION
                ,
                $OROC
                ,
                $ROctype
                ) ;

            $assessment[$newDMU]["DiffO"] = $assessment[$newDMU]["estTimeO"]
                - $DMU_Data[$newDMU]["Time"] ;
        }
        if ( $iOrientation == INPUT_ORIENTATION ||
            $iValidate == VALIDATE_BOTH ) {
            $assessment[$newDMU]["estTimeI"] = estimateTime( $newDMU
                ,

```

```

                $periodNewSOA[$soaPeriod],
                $Inputs
                ,
                $Outputs
                ,
                $iRTS
                ,
                $DMU_Data
                ,
                $soaPeriod
                ,
                INPUT_ORIENTATION
                ,
                $IROC
                ,
                $ROctype
                );

        $assessment[$newDMU]["DiffI"] = $assessment[$newDMU]["estTimeI"]
        - $DMU_Data[$newDMU]["Time"] ;
    }
}
}
tableHeader ("DMUs after $PredictBase($stopTime) Feasibility", "DMU",
    array("Available"      , "Status"      , "Not Agressive"  ,
          "Aggressive"    , "Est. OTime"   , "Delta OTime"   ,
          "Est. ITime"    , "Delta ITime"  ) );
$rowCount = 0 ;
foreach ( array_keys($assessment) as $reportedDMU ) {
    tableRow($DMU_Name[$reportedDMU] . "(" . $reportedDMU . ")",
        array(getStamp($DMU_Data[$reportedDMU]["Time"], $connection),
              $assessment[$reportedDMU]["conclusion"] .
              ":" .
              $reportStatus[$assessment[$reportedDMU]["conclusion"]],
              $assessment[$reportedDMU]["LowerScore"]
              ,
              $assessment[$reportedDMU]["UpperScore"]
              ,
              (( isset($assessment[$reportedDMU]["estTimeO"]) ) ?
                getStamp($assessment[$reportedDMU]["estTimeO"] ,
                          $connection
                          ) :
                "INF"
                ) ,
              $assessment[$reportedDMU]["DiffO"]
              ,
              (( isset($assessment[$reportedDMU]["estTimeI"]) ) ?
                getStamp($assessment[$reportedDMU]["estTimeI"] ,
                          $connection
                          ) :
                "INF"
                ) ,
              $assessment[$reportedDMU]["DiffI"]
              ),
        $rowCount , 0 );
    $rowCount++;
}
tableFooter ("");
}

if ( $bPredict ) {
    if ( ! $PredictBase ) {
        echo "ERROR: No Base set\n";
    } elseif ( $PredictPeriods ) {
        $referencePeriod = getPeriod($PredictBase, $connection) ;
        foreach ( $PredictPeriods as $TimeToPredict ) {
            $periodToPredict = $referencePeriod + $TimeToPredict ;
            $predictStamp = getStamp($periodToPredict, $connection );

            if ( isset($periodNewSOA[$referencePeriod]) ) {
                $Frontier = ForecastSOA( $referencePeriod, $periodToPredict,
                    $iOrientation , $iRTS
                    ,
                    $iValidate , $ROctype
                    ,
                    $periodNewSOA[$referencePeriod]
                    ,
                    $IsumStats[$referencePeriod]
                    ,
                    $OsumStats[$referencePeriod]
                    ,
                    $Inputs
                    , $Outputs
                    ,
                    $DMU_Data ) ;
                tableHeader("Forecast from " . $PredictBase . " to "
                    . $predictStamp , "", array(""));
                if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
                    echo " <tr>\n";
                }
            }
        }
    }
}

```



```

foreach ( array("Lower", "Upper") as $which ) {
    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "          <td valign=top>\n";
    }
    tableHeader($which
                ,
                "Product"
                ,
                array_merge( $Inputs, $Outputs ) );
    $rowCount = 0 ;
    foreach ( array_keys( $Frontier[$which] ) as $keyDMU ) {
        tableRow( $keyDMU , $Frontier[$which][$keyDMU],
                 $rowCount, 0 );
        $rowCount++;
    }
    tableFooter("");
    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "          </td>\n";
    }
}
if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
    echo "    </tr>\n";
}
}
}
tableFooter("");
}
}
}

function EvaluateForecast( $iTime      , $iOrientation, $iRTS      ,
                          $iValidate  , $ROctype    , $soaDMUs   ,
                          $newDMU     , $Istats     , $Ostats    ,
                          $Inputs     , $Outputs    , $DMUData   ,
                          $bTrace     ) {

    $functionName = "EvaluateForecast";
    logTrace($functionName, 100000200, "Entering...");

    $Lower = array();
    $Upper = array();

    $currTime = $iTime;
    $evalTime = $DMUData[$newDMU]["Time"] ;
    $rowCount = 0;

    if ( ! in_array( $newDMU, $soaDMUs ) ) {
        if ( $Ostats[0] != 1 || $Istats[0] != 1 ) {
            if ( $currTime < $evalTime ) {
                $Frontier = ForecastSOA( $currTime      , $evalTime      ,
                                       $iOrientation   , $iRTS           ,
                                       $iValidate      , $ROctype       ,
                                       $soaDMUs        , $Istats        ,
                                       $Ostats        , $Inputs        ,
                                       $Outputs        , $DMUData       ) ;

                //
                // Add to the upper and lower confidence intervals the currently
                // evaluated DMU.
                //
                $Lower=$Frontier["Lower"]           ;
                $Upper=$Frontier["Upper"]           ;
                $Lower[$newDMU]=$DMUData[$newDMU] ;
                $Upper[$newDMU]=$DMUData[$newDMU] ;

                //
                // Calculate the results
                //
                if ( $iOrientation == OUTPUT_ORIENTATION ) {
                    $LowerResult = CCR( $newDMU      , 1
                                       , $Inputs      , $Outputs      ,

```

```

        array_keys($Lower), $iRTS      ,
        $bTrace             , $Lower   );

    $UpperResult = CCR( $newDMU        , 1          ,
        $Inputs             , $Outputs    ,
        array_keys($Upper), $iRTS      ,
        $bTrace             , $Upper     );
} elseif ( $iOrientation == INPUT_ORIENTATION ) {
    $LowerResult = IO( $newDMU        , 1          ,
        $Inputs             , $Outputs    ,
        array_keys($Lower), $iRTS      ,
        $bTrace             , $Lower     );

    $UpperResult = IO( $newDMU        , 1          ,
        $Inputs             , $Outputs    ,
        array_keys($Upper), $iRTS      ,
        $bTrace             , $Upper     );
}

//
// Report what we did and where we are...
//
if ( $bTrace ) {
    //
    // Add prefix
    //
    tableHeader ( "Lower Bounds", "DMU",
        array_merge($Inputs, $Outputs));
    foreach ( array_keys($Lower) as $key ) {
        tableRow( $key, $Lower[$key], $rowCount, 0 );
        $rowCount++;
    }
    tableFooter("");
    tableHeader ( "Upper Bounds", "DMU",
        array_merge($Inputs, $Outputs));
    foreach ( array_keys($Upper) as $key ) {
        tableRow( $key, $Upper[$key], $rowCount, 0 );
        $rowCount++;
    }
    tableFooter("");

    reportMessage($currTime . ":" . $LowerResult["Theta"] . ":"
        . $UpperResult["Theta"] );
}

$Forecast["LowerScore"] = $LowerResult["Theta"] ;
$Forecast["UpperScore"] = $UpperResult["Theta"] ;

logTrace($functionName, 100000210, " ->LowerScore ["
    . $LowerResult["Theta"] . "]" );
if ( ! $LowerResult["Theta"] || !isset($LowerResult["Theta"]) ) {
    $Forecast["Lower"] = -20 ;
    logTrace($functionName, 100000211, " -->Broken");
} elseif ( ( $iOrientation == OUTPUT_ORIENTATION &&
    $LowerResult["Theta"] <= ( 1 + CLOSE_ENOUGH ) ) ||
    ( $iOrientation == INPUT_ORIENTATION &&
    $LowerResult["Theta"] >= ( 1 - CLOSE_ENOUGH ) ) ) {
    $Forecast["Lower"] = 1 ;
    logTrace($functionName, 100000212, " -->Good");
} else {
    $Forecast["Lower"] = 0 ;
    logTrace($functionName, 100000214, " -->Not Good");
}

logTrace($functionName, 100000210, " ->UpperTheta ["
    . $UpperResult["Theta"] . "]" );
if ( ! $UpperResult["Theta"] || !isset($UpperResult["Theta"]) ) {
    $Forecast["Upper"] = -20 ;

```

```

        logTrace($functionName, 100000211, " -->Broken");
    } elseif ( ( $iOrientation == OUTPUT_ORIENTATION &&
        $UpperResult["Theta"] <= ( 1 + CLOSE_ENOUGH ) ) ||
        ( $iOrientation == INPUT_ORIENTATION &&
        $UpperResult["Theta"] >= ( 1 - CLOSE_ENOUGH ) ) ) {
        logTrace($functionName, 100000212, " -->NotGood");

        $Forecast["Upper"] = 0 ;
    } else {
        $Forecast["Upper"] = 1 ;
        logTrace($functionName, 100000213, " -->Good");
    }
} else {
    $Forecast["Lower"] = $Forecast["Upper"] = -1;
}
} else {
    logTrace ( $functionName, 3, "          No stat information" );
    $Forecast["Lower"] = $Forecast["Upper"] = -5 ;
}
} else {
    $Forecast["Lower"] = $Forecast["Upper"] = -10 ;
}
return($Forecast);
logTrace($functionName, 100000299, "Leaving...");
}

//
// buildProxyDMU
//
// This function takes the DMU identified, confirms if it is efficient and
// if not returns a proxy dmu based on the result matrix.
//
function buildProxyDMU ( $DMU          , $DMUData          ,
                        $iOrientation  , $Inputs          ,
                        $iOutputs      , $ResultArray     ) {

    //
    // Set the result row to be equal to the dmu in question
    //
    $ResultRow = $ResultArray[$DMU] ;

    if ( ( $iOrientation == OUTPUT_ORIENTATION &&
        $ResultArray[$DMU]["Theta"] >= ( 1 + CLOSE_ENOUGH ) ) ||
        ( $iOrientation == INPUT_ORIENTATION &&
        $ResultArray[$DMU]["Theta"] <= ( 1 - CLOSE_ENOUGH ) ) ) {
        foreach ( array_keys($ResultRow) as $ArrayKey ) {
            if ( substr_count( $ArrayKey, "Lambda" ) ) {
                if ( $ResultRow[$ArrayKey] > CLOSE_ENOUGH ) {
                    $ReferenceDMU = substr( $ArrayKey,
                        ( strpos( $ArrayKey, "-" ) + 1 ) );
                    if ( $iOrientation == OUTPUT_ORIENTATION ) {
                        foreach ( $Outputs as $Factor ) {
                            $VirtualRow[$Factor] = $ResultRow[$ArrayKey] *
                                $DMUData[$ReferenceDMU][$Factor] ;
                        }
                    }
                    foreach ( $Inputs as $Factor ) {
                        $VirtualRow[$Factor] = $DMUData[$DMU][$Factor] ;
                    }
                }
            }
            if ( $iOrientation == INPUT_ORIENTATION ) {
                foreach ( $Inputs as $Factor ) {
                    $VirtualRow[$Factor] = $ResultRow[$ArrayKey] *
                        $DMUData[$ReferenceDMU][$Factor] ;
                }
                foreach ( $Outputs as $Factor ) {
                    $VirtualRow[$Factor] = $DMUData[$DMU][$Factor] ;
                }
            }
        }
    }
}
}

```

```

        } else {
            $VirtualRow[$Factor] = $DMUData[$DMU][$Factor] ;
        }
    }
} else {
    $VirtualRow = $DMUData[$DMU] ;
}
}

function estimateTime( $DMU          , $soaDMUs      , $Inputs   ,
                      $Outputs      , $iRTS      , $DMU_Data ,
                      $iTime         , $iOrientation , $ROC      ,
                      $rocType       ) {

    $functionName = "estimateTime";
    $Result       = array();

    logTrace($functionName, 2100, "Entering function...");
    logTrace($functionName, 2101, " DMU [$DMU]");
    foreach ( $soaDMUs as $element ) {
        logTrace($functionName, 2112, " DMU [$element] ");
    }
    foreach ( $Inputs as $input ) {
        logTrace($functionName, 2113, " Input [$input]");
    }
    foreach ( $Outputs as $output ) {
        logTrace($functionName, 2114, " Output [$output]" );
    }
    logTrace($functionName, 2115, " iRTS [$iRTS] Time [$iTime] " .
        "iOrientation [$iOrientation] ROC [$ROC]");

    if ( $iOrientation == OUTPUT_ORIENTATION ) {
        logTrace($functionName, 2102, " Output Orientation, calculating...");
        $Result = CCR( $DMU          , 1          ,
                      $Inputs      , $Outputs   ,
                      $soaDMUs     , $iRTS      ,
                      0            , $DMU_Data  ) ;
    } elseif ( $iOrientation == INPUT_ORIENTATION ) {
        logTrace($functionName, 2103, " Input Orientation, calculating...");
        $Result = IO( $DMU          , 1          ,
                     $Inputs      , $Outputs   ,
                     $soaDMUs     , $iRTS      ,
                     0            , $DMU_Data  ) ;
    }
    logTrace($functionName, 2112, " DMU Time [" .
        $DMU_Data[$DMU]['Time'] . "]);

    if ( ! isset($Result["Theta"]) ) {
        logTrace($functionName, 2120, " Theta not calculated...");
    } else {
        logTrace($functionName, 2111, " Theta [" . $Result["Theta"] . "]);
        $logTheta = log($Result["Theta"]) ;
        logTrace($functionName, 2107, " logTheta [$logTheta]");

        $logROC = log($ROC) ;
        logTrace($functionName, 2108, " logROC [$logROC]");

        $deltaTime = $logTheta / $logROC ;
        logTrace($functionName, 2106, " Estimate intro delta t [$deltaTime]");

        $rocType = ROC_COMPOSITE_TIME ;
        if ( $rocType == ROC_COMPOSITE_TIME ) {
            logTrace($functionName, 2117, " -> Composite time selected");
            $lambdaTimes = 0;
            $lambdaTotal = 0;
            foreach ( array_keys( $Result ) as $key ) {
                logTrace($functionName, 2118, " --> Key [" . $key . "]);
            }

```

```

//
// Only count the parameters that are Lambdas
//
if ( substr_count($key,"Lambda-") ) {
//
// Only count if we are any significant value
//
logTrace($functionName, 2119, " ---> Is a lambda");
if ( $Result[$key] > CLOSE_ENOUGH ) {
logTrace($functionName, 2120, " ----> Key [" . $key
. "] is [" . $Result[$key] . "]);
$obj = substr($key, (strpos($key,"-")+1) );
$lambdaTimes += ( $DMU_Data[$obj]["Time"] * $Result[$key] ) ;
logTrace($functionName, 2121, " ----> LambdaTime adding [" .
( $DMU_Data[$obj]["Time"] * $Result[$key] ) . "]);
$lambdaTotal += $Result[$key] ;
logTrace($functionName, 2121, " ----> Lambda adding [" .
$Result[$key] . "]);
}
}
}

logTrace($functionName, 2122, " ---> LambdaTimes [" . $lambdaTimes
. "]);
logTrace($functionName, 2123, " ---> LambdaTotal [" . $lambdaTotal
. "]);

$timeOfIntro = ( $iOrientation == OUTPUT_ORIENTATION ) ?
(( ( 0 - $deltaTime ) * $lambdaTotal) +
($lambdaTimes)) / $lambdaTotal :
(( ( 0 + $deltaTime ) * $lambdaTotal) +
($lambdaTimes)) / $lambdaTotal ;
} else {
$timeOfIntro = ( $iOrientation == OUTPUT_ORIENTATION ) ? ( $iTime -
$deltaTime):
( $iTime +
$deltaTime);
}

$returnValue = $timeOfIntro ;

logTrace($functionName, 2116, " Estimate time [" . $returnValue . "]);
logTrace($functionName, 2110, "Leaving function...");
return($returnValue);
}
}
?>

```

DEAFunctions.php

```

<?php
// $Id: DEAFunctions.php,v 1.17 2004/07/04 20:35:33 oli Exp $
//
// Description : This set of routines solves DEA problems with the use of
//              : glpk-php extension
//
// Author      : Lane Inman
//              : Portland State University
//
define("CLOSE_ENOUGH", 0.000000000001) ;// At this point it is close enough
// to be 0 ;
define("EPSILON"      , 0.0000000000000001);// this is epsilon...

define("COLOR[0]"    , "white" );
define("COLOR[1]"    , "grey" );

```

```

//
// Returns To Scale : RTS
//
define("RTS_CRSS", 0 ) ; // Constant Returns to Scale
define("RTS_VRS", 1 ) ; // Variable Returns To Scale
define("RTS_DRS", 2 ) ; // Decreasing Returns to Scale
define("RTS_IRS", 3 ) ; // Increasing Returns to Scale

//
// Orientation
//
define("INPUT_ORIENTATION" , 0 ) ; // Input Oriented
define("OUTPUT_ORIENTATION", 1 ) ; // Output Oriented
function orientationString( $iOrientation ) {
    if ( $iOrientation == OUTPUT_ORIENTATION ) { return("OUTPUT"); }
    elseif ( $iOrientation == INPUT_ORIENTATION ) { return("INPUT"); }
}

//
// Display Options
//
define("DISPLAY_HTML" , 0 ) ; // Display HTML
define("DISPLAY_TEXT" , 1 ) ; // Display TEXT

//
// This function solves CCR DEA
//
function CCR ( $Observation, $bSuper, $Inputs, $Outputs, $DMUs,
              $iRTS, $iDebug, $DMU_Data ) {

    $numberOfColumns=count($DMUs)+1;
    $labels = $objective = $rowLabels = $restraint = $matrix = $labels = array();

    $iDebug = 0;
    if ( $iDebug ) {
        echo "DMU: $Observation\n";
    }
    if ( $iRTS ) {
        $numberOfRows=count($Outputs)+count($Inputs)+1;
    } else {
        $numberOfRows=count($Outputs)+count($Inputs);
    }

    foreach ( $DMUs as $DMU ) {
        $labels=array_merge($labels, array("Lambda-$DMU"));
    }

    $labels=array_merge($labels, array("Theta"));

    $res = glpk_create( "problem", $numberOfRows, $numberOfColumns,
                      LPX_MAX, LPX_LP, "Theta" );

    $Counter=1;
    foreach ( $DMUs as $DMU ) {
        if ( ($bSuper == 1) && ( $DMU == $Observation ) ) {
            glpk_set_col_bound($res, $Counter, LPX_FX, 0);
        } else {
            glpk_set_col_bound($res, $Counter, LPX_LO, 0,0);
        }
        $objective=array_merge($objective,array(0));
        $Counter++;
    }
    #
    # Theta
    #
    $objective=array_merge($objective,array(1));

```

```

glpk_set_col_bound($res, $Counter, LPX_LO, 0,0);

glpk_set_col_names($res, $labels);

glpk_set_obj_coeffs( $res,$objective);

$Counter=1;

foreach ( $Inputs as $Input ) {
  $rowLabels=array_merge($rowLabels,array($Input));
}
foreach ( $Outputs as $Output ) {
  $rowLabels=array_merge($rowLabels,array($Output));
}
if ( $iRTS == RTS_VRS ) {
  $rowLabels=array_merge($rowLabels,array("VRS"));
} else if ( $iRTS == RTS_DRS ) {
  $rowLabels=array_merge($rowLabels, array("DRS"));
} else if ( $iRTS == RTS_IRS ) {
  $rowLabels=array_merge($rowLabels, array("IRS"));
}

glpk_set_row_names( $res, $rowLabels);

foreach ( $Inputs as $Input ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint,array($DMU_Data["$DMU"][$Input]));
  }
  $restraint=array_merge($restraint,0);
  glpk_set_row_bound( $res, $Counter, LPX_UP, $DMU_Data[$Observation][$Input]);

  $matrix=array_merge($matrix,$restraint);
  $restraint=array();
  $Counter++;
}

foreach ( $Outputs as $Output ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint, array(-$DMU_Data["$DMU"][$Output]));
  }
  $restraint=array_merge($restraint, array($DMU_Data[$Observation][$Output]));
  $matrix=array_merge($matrix,$restraint);
  glpk_set_row_bound( $res, $Counter, LPX_UP, 0);
  $restraint=array();
  $Counter++;
}

if ( $iRTS ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint,array(1));
  }
  $restraint=array_merge($restraint, array(0));
  $matrix=array_merge($matrix, $restraint);
  if ( $iRTS == RTS_VRS ) {
    glpk_set_row_bound( $res, $Counter, LPX_FX, 1);
  } elseif ( $iRTS == RTS_DRS ) {
    glpk_set_row_bound( $res, $Counter, LPX_UP, 1);
  } elseif ( $iRTS == RTS_IRS ) {
    glpk_set_row_bound( $res, $Counter, LPX_LO, 1 );
  }
  $restraint=array();
  $Counter++;
}

glpk_set_matrix($res, $matrix);

$result = glpk_solve( $res );

```

```

$result = glpk_solve( $res );
if ( $iDebug == 1 ) {
    switch ( $result->result )
    {
        case LPX_E_OK      : print( "Problem successfully solved.<br>" ); break;
        case LPX_E_FAULT  : print( "Faulty problem.  Solution search not started.<br>\n" );
break;
        case LPX_E_OBJLL  : print( "Objective function continually decreasing.<br>\n" );
break;
        case LPX_E_OBJUL  : print( "Objective function continually increasing.<br>\n" );
break;
        case LPX_E_ITLIM  : print( "Iterations limit exceeded.<br>\n" ); break;
        case LPX_E_TMLIM  : print( "Time limit exceeded.<br>\n" ); break;
        case LPX_E_SING   : print( "Solver failuer -- singular or ill-conditioned.<br>\n" );
break;
        case LPX_E_NOPFS  : print( "No primal feasible solution.<br>\n" ); break;
        case LPX_E_NODFS  : print( "No dual feasible solution.<br>\n" ); break;
    }
    switch ( $result->status )
    {
        case LPX_OPT      : print( "Optimal solution found.<br>\n" ); break;
        case LPX_FEAS     : print( "Solution is feasible.<br>\n" ); break;
        case LPX_INFEAS   : print( "Solution is infeasible.<br>\n" ); break;
        case LPX_NOFEAS   : print( "Problem has no feasible solution.<br>\n" ); break;
        case LPX_UNBND    : print( "Unbounded solution.<br>\n" ); break;
        case LPX_UNDEF    : print( "Solution status is undefined.<br>\n" ); break;
    }
}

//

foreach ( $result->cols as $col )
{
    //
    // This is here to get rid of bogus small entries, I am not sure if it
    // is a good idea, but I am doing it to clean up my tables.
    //
    //if ( $col->primal < EPSILON ) {
    //    $return_array[$col->name] = 0;
    //} else {
    //    $return_array[$col->name]=$col->primal ;
    //}
}
return( $return_array );
glpk_delete( $res );
}

function IO ( $Observation, $bSuper, $Inputs, $Outputs, $DMUs,
             $iRTS, $iDebug, $DMU_Data ) {

    $labels = $objective = $rowLabels = $restraint = $matrix = $labels = array();

    if ( $iDebug ) {
        echo "DMU: $Observation\n";
    }
    $numberOfCols=count($DMUs)+1;
    if ( $iRTS ) {
        $numberOfRows=count($Outputs)+count($Inputs)+1;
    } else {
        $numberOfRows=count($Outputs)+count($Inputs);
    }

    foreach ( $DMUs as $DMU ) {
        $labels=array_merge($labels, array("Lambda-$DMU"));
    }

    $labels=array_merge($labels, array("Theta"));

    $res = glpk_create( "problem", $numberOfRows, $numberOfCols,
                      LPX_MIN, LPX_LP, "Theta" );

```



```

$Counter=1;
foreach ( $DMUs as $DMU ) {
  if ( $bSuper && ( $DMU == $Observation ) ) {
    glpk_set_col_bound($res, $Counter, LPX_FX, 0);
  } else {
    glpk_set_col_bound($res, $Counter, LPX_LO, 0,0);
  }
  $objective=array_merge($objective,array(0));
  $Counter++;
}
$objective=array_merge($objective,array(1));

glpk_set_col_names($res, $labels);
glpk_set_col_bound($res, $Counter, LPX_LO, 0,0);

glpk_set_obj_coeffs( $res,$objective);

$Counter=1;

foreach ( $Inputs as $Input ) {
  $rowLabels=array_merge($rowLabels,array($Input));
}
foreach ( $Outputs as $Output ) {
  $rowLabels=array_merge($rowLabels,array($Output));
}
if ( $iRTS == RTS_VRS ) {
  $rowLabels=array_merge($rowLabels,array("VRS"));
} else if ( $iRTS == RTS_DRS ) {
  $rowLabels=array_merge($rowLabels, array("DRS"));
} else if ( $iRTS == RTS_IRS ) {
  $rowLabels=array_merge($rowLabels, array("IRS"));
}

glpk_set_row_names( $res, $rowLabels);
foreach ( $Inputs as $Input ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint, array(-$DMU_Data["$DMU"][$Input]));
  }
  $restraint=array_merge($restraint, array($DMU_Data[$Observation][$Input]));
  $matrix=array_merge($matrix,$restraint);
  glpk_set_row_bound( $res, $Counter, LPX_LO, 0);
  $restraint=array();
  $Counter++;
}

foreach ( $Outputs as $Output ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint,array($DMU_Data["$DMU"][$Output]));
  }
  $restraint=array_merge($restraint,0);
  glpk_set_row_bound( $res, $Counter, LPX_LO, $DMU_Data[$Observation][$Output]);

  $matrix=array_merge($matrix,$restraint);
  $restraint=array();
  $Counter++;
}

if ( $iRTS ) {
  foreach ( $DMUs as $DMU ) {
    $restraint=array_merge($restraint,array(1));
  }
  $restraint=array_merge($restraint, array(0));
  $matrix=array_merge($matrix, $restraint);
  if ( $iRTS == RTS_VRS ) {
    glpk_set_row_bound( $res, $Counter, LPX_FX, 1);
  } elseif ( $iRTS == RTS_DRS ) {
    glpk_set_row_bound( $res, $Counter, LPX_UP, 1);
  }
}

```

```

    } elseif ( $iRTS == RTS_IRS ) {
        glpk_set_row_bound( $res, $Counter, LPX_LO, 1 );
    }
    $Counter++;
}

glpk_set_matrix($res, $matrix);
$result = glpk_solve( $res );

$result = glpk_solve( $res );
if ( $iDebug == 1 ) {
    switch ( $result->result ) {
        case LPX_E_OK      : print( "Problem successfully solved.<br>" ); break;
        case LPX_E_FAULT  : print( "Faulty problem.  Solution search not started.<br>\n" );
break;
        case LPX_E_OBJLL  : print( "Objective function continually decreasing.<br>\n" );
break;
        case LPX_E_OBJJUL : print( "Objective function continually increasing.<br>\n" );
break;
        case LPX_E_ITLIM  : print( "Iterations limit exceeded.<br>\n" ); break;
        case LPX_E_TMLIM  : print( "Time limit exceeded.<br>\n" ); break;
        case LPX_E_SING   : print( "Solver failuer -- singular or ill-conditioned.<br>\n" );
break;
        case LPX_E_NOPFS  : print( "No primal feasible solution.<br>\n" ); break;
        case LPX_E_NODFS  : print( "No dual feasible solution.<br>\n" ); break;
    }
    switch ( $result->status ) {
        case LPX_OPT      : print("Optimal solution found.<br>\n"); break;
        case LPX_FEAS     : print("Solution is feasible.<br>\n"); break;
        case LPX_INFEAS   : print("Solution is infeasible.<br>\n"); break;
        case LPX_NOFEAS   : print("Problem has no feasible solution.<br>\n"); break;
        case LPX_UNBND    : print( "Unbounded solution.<br>\n" ); break;
        case LPX_UNDEF    : print( "Solution status is undefined.<br>\n" ); break;
    }
}

//

if ( $result->status == LPX_OPT ) {
    foreach ( $result->cols as $col ) {
        //
        // This is here to get rid of bogus small entries, I am not sure if it
        // is a good idea, but I am doing it to clean up my tables.
        //
        // if ( $col->primal < EPSILON ) {
        //     $return_array[$col->name] = 0;
        // } else {
        //     $return_array[$col->name]=$col->primal ;
        // }
    }
}
return( isset($return_array) ? $return_array : array());
glpk_delete( $res );
}

function ShowStandardResults ( $iOrientation, $DMUs, $Solution, $Lambdas, $Prog ) {
    $rowColors=array("silver","white");
    $effColor="yellow";

    echo "<table border=0 cellspacing=0 >\n"
        . " <tr bgcolor=\"black\">\n"
        . " <td></td>\n";
    foreach ( $DMUs as $DMU ) {
        if ( $Lambdas["$DMU"] > 0 ) {
            echo " <td align=center><font style=\"color:#FFFFFF;\n"
                . " font-family:Arial,Helvetica,Sans-Serif\">\n"
                . " <b>&lambda<sub><font size=-1>$DMU</font></sub></b></font>\n"
                . " </td>\n";
        }
    }
}

```

```

}
echo " <td align=center>\n"
. " <font style=\"color:#FFFFFF;\n"
. " font-family:Arial,Helvetica,Sans-Serif\">\n"
. " <b>&theta;</b>\n"
. " </td>\n";
if ( $Prog ) {
echo " <td align=center>\n"
. " <font style=\"color:#FFFFFF;\n"
. " font-family:Arial,Helvetica,Sans-Serif\">\n";
if ( $iOrientation ) {
echo " <b>&beta;</b>\n";
} else {
echo " <b>&gamma;</b>\n";
}
echo " </td>\n";
}
}

echo " </tr>\n";

foreach ( $DMUs as $DMU ) {
if ( $iOrientation && $Solution["$DMU"]['Theta'] <= 1.00000000000001 ) {
$rowColor=$effColor ;
} else if ( ! $iOrientation && $Solution["$DMU"]['Theta'] >= 0.99999999999999 ) {
$rowColor=$effColor ;
} else if ( $rowColorIndex == 0 ) {
$rowColorIndex=1 ;
$rowColor=$rowColors[$rowColorIndex] ;
} else {
$rowColorIndex=0 ;
$rowColor=$rowColors[$rowColorIndex] ;
}

echo " <tr bgcolor=$rowColor>\n";
echo " <td>$DMU</td>\n";
foreach ( $DMUs as $Lambda ) {
if ( $Lambdas[$Lambda] > 0 ) {
echo " <td align=center>.$Solution["$DMU"]["Lambda-$Lambda"]."</td>\n";
}
}
echo " <td align=center>{$Solution["$DMU"]['Theta']}</td>\n";
if ( $Prog["$DMU"] ) {
echo " <td align=center>{$Prog["$DMU"]}</td>\n";
}
}
echo " </tr>\n";
}
echo "</table>";
}

//
// ShowDEAResults
//
// This function displays the results of a DEA analysis.
//
// Variables : Format - 0 text
// : - 1 html tabular
// : iOrientation - is it an output model?
// : DMUs - The list of DMUs to print out;
// : Solution- The solution to printout
//
//

function ShowDEAResults ($iFormat, $iOrientation, $lsDMUs, $lrlSolution) {

$color[0] = "white" ;
$color[1] = "silver";
$keysToPrint = array() ;

```

```

foreach ( $lrlSolution as $resultLine ) {
  foreach ( array_keys($resultLine) as $key ) {
    if ( $resultLine[$key] > 0.00 && $key != "Theta" ) {
      $keysToPrint[$key]=1;
    }
  }
}

//
// Header Information - We should probably do better than this, but I am
//                       - not so worried about these forms... A "prep-table"
//                       - Function would probabably be better.
//
if ( $iFormat == DISPLAY_TEXT ) {
  echo "DMU : ";
  if ( $iOrientation == INPUT_ORIENTATION ) {
    echo "Theta : ";
  } else if ( $iOrientation == OUTPUT_ORIENTATION ) {
    echo "Phi : ";
  }
  foreach ( array_keys($keysToPrint) as $key ) {
    if ( $key != "Theta" ) {
      echo "$key : " ;
    }
  }
  echo "\n";
}

if ( $iFormat == DISPLAY_HTML ) {
  echo "<table width=100% cellpadding=0 cellspacing=0 border=0>\n";
  echo "  <tr bgcolor=" . TABLE_HEADER_BG . " style='font-family:"
    . FONT_STYLE . ";color:white'> \n";
  echo "    <td align=center>DMU</td>\n";

  //
  // The efficiency coefficient is different based on the model being used.
  // Here we use phi if it is output and theta if it is input
  //
  if ( $iOrientation == INPUT_ORIENTATION ) {
    echo "      <td align=center>&theta;</td>\n";
  } else if ( $iOrientation == OUTPUT_ORIENTATION ) {
    echo "      <td align=center>&phi;</td>\n";
  }

  //
  // Converting header to lambda subscript
  //
  foreach ( array_keys($keysToPrint) as $Key ) {
    $stringToDisplay = "&lambda;<sub><font size=-1>" .
      substr($Key, (strpos($Key, "-")+1))
      . "</font></sub>";
    echo "      <td align=center>" . $stringToDisplay . "</td>\n";
  }

  echo "    </tr>\n";
}

$colorCount = 0;
asort($lsDMUs);
foreach ( $lsDMUs as $DMU ) {
  if ( ( $lrlSolution["$DMU"]['Theta'] >= ( 1 + EPSILON ) &&
        $iOrientation == INPUT_ORIENTATION ) ||
        ( $lrlSolution["$DMU"]['Theta'] <= ( 1 - EPSILON ) &&
        $iOrientation == OUTPUT_ORIENTATION ) ) {
    ( $iFormat == DISPLAY_HTML ) ?
      $printString = "      <tr bgcolor=yellow>\n" :
      $printString = "      ";
  } else {

```

```

        ( $iFormat == DISPLAY_HTML ) ?
        $printString = "      <tr bgcolor='" . $color[$colorCount % 2]
          . "'>\n" :
        $printString = "";
        $colorCount++;
    }
    ( $iFormat == DISPLAY_HTML ) ?
    $printString .= "      <td align=center>" . $DMU . "</td>\n"
      . "      <td align=center>" . $lrlSolution["$DMU"]['Theta']
      . "</td>\n" :
    $printString .= $DMU . ":" . $lrlSolution["$DMU"]['Theta'] . ":" ;

    foreach ( array_keys($keysToPrint) as $key ) {
        ( $iFormat == DISPLAY_HTML ) ?
        $printString .= "      <td align=center>"
          . $lrlSolution["$DMU"][$key]
          . "</td>\n" :
        $printString .= ":" . $lrlSolution["$DMU"][$key] ;
    }
    ( $iFormat == DISPLAY_HTML ) ? $printString .= "    </tr>\n" :
    $printString .= "\n";
    echo $printString ;
}
if ( $iFormat == DISPLAY_HTML ) { echo "</table>\n"; }
}

?>

```

MathFunctions.php

```

<?php
//
// Lane Inman
// Engineering and Technology Management Department
// Portland State University
//
// $Id: MathFunctions.php,v 1.19 2004/06/16 04:36:16 oli Exp $
//

define("ROC_COMPOSITE_TIME", 1 ) ; // Defined as composite time for event
define("ROC_CURRENT_TIME" , 2 ) ; // Defined as time since current

define("VALIDATE_INPUT" , 0 ) ; // Validate against input projection
define("VALIDATE_OUTPUT" , 1 ) ; // Validate against output projection
define("VALIDATE_BOTH" , 2 ) ; // Validate against input and output

define("FORECAST_ABOVE_LOWER" , 1 ) ;
define("FORECAST_BELOW_UPPER" , 2 ) ;
define("FORECAST_GOOD" , 3 ) ;
define("FORECAST_NOT_COVERED" , 4 ) ;
define("FORECAST_ERROR" , 5 ) ;

//
// Function : FindStats
//
// Description : This calculates the mean, standard deviation, and the 95%
// : confidence interval of an array.
//
// Variables : Data - array of data.
//
// Return Value : This is an array with the following:
// :
// : Mean of the values provided;
// : Standard deviation ;
// : Confidence interval ;
//

```

```

function FindStats ( $Data, $iOrientation ) {

    $functionName = "FindStats";

    logTrace ( $functionName, 30000000, "Entering...");
    logTrace ( $functionName, 30000001, " -> Orientation: " . $iOrientation);
    //
    // Assign Variables
    //
    $totalSum = $totalCount = $totalSqrs = 0;

    //
    // Don't continue if there is less than one
    //
    if ( count($Data) > 1 ) {
        logTrace( $functionName, 30000002, " -> count(Data) [" . count($Data)
            . "]" );
        foreach ( array_keys($Data) as $key ) {
            if ( ( $Data[$key] > ( 1 + CLOSE_ENOUGH ) &&
                $iOrientation == OUTPUT_ORIENTATION ) ||
                ( $Data[$key] < ( 1 - CLOSE_ENOUGH ) &&
                $iOrientation == INPUT_ORIENTATION && $Data[$key] != 1 ) ) {
                logTrace($functionName, 24, " -->DMU: " . $key . " Data: " .
                    $Data[$key] . " - recording " );
                $totalSum += $Data[$key] ;
                $totalCount += 1 ;
                $totalSqrs += pow($Data[$key], 2);
            }
        }
        logTrace($functionName, 30000003, " -> SUM : " . $totalSum );
        logTrace($functionName, 30000004, " -> totalCount: " . $totalCount);
        logTrace($functionName, 30000005, " -> totalSqrs : " . $totalSqrs );

        if ( $totalCount > 2 ) {
            $meanScore = $totalSum / $totalCount ;
            $stdDev = pow( ( $totalCount * $totalSqrs - pow($totalSum,2) ) /
                ( $totalCount * ( $totalCount - 1 ) ), .5);
            $confInt = $stdDev / pow($totalCount, .5) * 1.96 ;
        } else {
            $meanScore = 1 ; $stdDev = 0 ; $confInt = 0 ;
        }
    } else {
        $meanScore = 1 ; $stdDev = 0 ; $confInt = 0 ;
    }

    logTrace($functionName, 24, "MeanScore: " . $meanScore .
        " Standard Deviation: " . $stdDev . " Confidence Interval: " .
        $confInt );
    logTrace($functionName, 25, "Exiting");;
    $resultArray = array( $meanScore, $stdDev, $confInt );
    return $resultArray ;
}

//
// Analyze Progress
//
// The purpose of this function is to analyze the overall progress of
// a DMU with respect to the frontier over a period of time...
//
// It returns the rate of change.
//
function AnalyzeProgress ( $DMU, $Time, $iOrientation, $DMUData,
    $resultArray, $DMUs , $timeMode, $iRTS ) {

    $functionName = "AnalyzeProgress";
    $LambdaTotal = 0.0 ;
    $bDone = 0 ;

```

```

logTrace ( $functionName, 40, "DMU: " . $DMU . " Time: " .
          $Time . " Orientation: " . $iOrientation );

//
// If the reference behavior is itself then no time has passed and we are
// done;
//
$timePassed = 0 ;

if ( $timeMode == ROC_COMPOSITE_TIME ) {
  if ( $resultArray["Lambda-$DMU"] > CLOSE_ENOUGH ) {
    $timePassed = 0 ;
    $bDone = 1 ;
  } else {
    foreach ( array_keys( $resultArray ) as $key ) {
      //
      // Only count the parameters that are Lambdas
      //
      if ( substr_count($key,"Lambda-") ) {
        //
        // Only count if we are any significant value
        //
        if ( $resultArray[$key] > CLOSE_ENOUGH && $bDone == 0 ) {
          $obj = substr($key, (strpos($key,"-")+1) );
          if ( $DMUData[$obj]["Time"] > $DMUData[$DMU]["Time"] ) {
            $timePassed += ( $DMUData[$obj]["Time"] -
                          $DMUData[$DMU]["Time"] ) *
                          $resultArray[$key] ;
          }
          $LambdaTotal += $resultArray[$key] ;
        }
      }
    }
  }
} elseif ( $timeMode == ROC_CURRENT_TIME ) {
  logTrace($functionName, 41, "Current time setting time to " .
          $Time . " - " . $DMUData[$DMU]["Time"] );
  $timePassed = $Time - $DMUData[$DMU]["Time"] ;
  logTrace($functionName, 42, "Current time set to " . $timePassed );
}

if ( ! $bDone &&
    ( $iOrientation == OUTPUT_ORIENTATION &&
      $resultArray["Theta"] >= ( 1 + CLOSE_ENOUGH ) ) ||
    ( $iOrientation == INPUT_ORIENTATION &&
      $resultArray["Theta"] <= ( 1 - CLOSE_ENOUGH ) ) ) {
  if ( $timePassed > 0 ) {
    //
    // This is needed in CRS because we are not guaranteed that
    // lambdas sum to one. in VRS however they do so if zero is
    // assigned to anything we can assume that it is a needed
    // divisor... If we are using not current time
    //
    if ( $iRTS == RTS_CRIS && $timeMode != ROC_CURRENT_TIME ) {
      logTrace($functionName, 43, "CRS and not CURRENT time so -" .
              " changing " . $timePassed .
              " using " . $LambdaTotal );
      $timePassed = $timePassed / $LambdaTotal ;
      logTrace($functionName, 44, "New Time - " . $timePassed );
    }
    $rateOfChange = pow($resultArray["Theta"],(1/$timePassed));
    logTrace($functionName, 45, $DMU . ":Passed:" . $timePassed . ":" .
            "eff:" . $resultArray["Theta"] . ":" .
            "ROC:" . $rateOfChange );
  } else {
    $rateOfChange = 1;
  }
} else {

```

```

        $rateOfChange = 1;
    }
    logTrace($functionName, 46, "ROC: $rateOfChange");
    return ( $rateOfChange );
}

function ForecastSOA ( $currTime      , $evalTime      , $iOrientation ,
                      $iRTS          , $iValidate     , $ROctype      ,
                      $soaDMUs       , $Istats        , $Ostats       ,
                      $Inputs         , $Outputs       , $DMUData      ) {

    $functionName = "ForecastSOA" ;

    // echo "TIMES: $currTime : $evalTime \n";
    // echo "Multiplier: $Ostats[0] : $Ostats[2] \n";

    ( $evalTime > $currTime ) ? $deltaTime = $evalTime - $currTime :
        $deltaTime = 0 ;

    foreach ( $soaDMUs as $DMU ) {
        //
        // If our time is current, then we keep deltaTime the same
        // otherwise, we assume that it is Composite Time.
        //
        ( $ROctype == ROC_CURRENT_TIME ) ? $deltaTime = $deltaTime :
            $deltaTime = $evalTime - $DMUData[$DMU]["Time"] ;
        logTrace( $functionName, 90, "TIMES: $currTime : " .
            $DMUData[$DMU]["Time"] );
        logTrace( $functionName, 91, "$DMU DeltaTime: $deltaTime " );
        logTrace( $functionName, 92, "$DMU ROC Lower Multi: " .
            pow(( $Istats[0] + $Istats[2] ), $deltaTime) );
        logTrace( $functionName, 93, "$DMU ROC Upper Multi: " .
            pow(( $Istats[0] - $Istats[2] ), $deltaTime) );
        logTrace( $functionName, 94, "$DMU ROC Upper Multi: " .
            pow(( $Ostats[0] + $Ostats[2] ), $deltaTime) );
        logTrace( $functionName, 95, "$DMU ROC Lower Multi: " .
            pow(( $Ostats[0] - $Ostats[2] ), $deltaTime) );
        foreach ( $Inputs as $Factor ) {
            if ( $iValidate == VALIDATE_INPUT ||
                $iValidate == VALIDATE_BOTH ) {

                $Lower["${DMU}I"][$Factor] = $DMUData[$DMU][$Factor] *
                    pow(( $Istats[0] + $Istats[2] ),
                        $deltaTime );
                $Upper["${DMU}I"][$Factor] = $DMUData[$DMU][$Factor] *
                    pow(( $Istats[0] - $Istats[2] ),
                        $deltaTime );
            }
            if ( $iValidate == VALIDATE_OUTPUT ||
                $iValidate == VALIDATE_BOTH ) {
                $Lower["${DMU}O"][$Factor] = $DMUData[$DMU][$Factor] ;
                $Upper["${DMU}O"][$Factor] = $DMUData[$DMU][$Factor] ;
            }
        }
    }

    foreach ( $Outputs as $Factor ) {
        if ( $iValidate == VALIDATE_OUTPUT ||
            $iValidate == VALIDATE_BOTH ) {
            $Lower["${DMU}O"][$Factor] = $DMUData[$DMU][$Factor] *
                pow(( $Ostats[0] - $Ostats[2] ),
                    $deltaTime);
            $Upper["${DMU}O"][$Factor] = $DMUData[$DMU][$Factor] *
                pow(( $Ostats[0] + $Ostats[2] ),
                    $deltaTime);
        }
        if ( $iValidate == VALIDATE_INPUT ||
            $iValidate == VALIDATE_BOTH ) {
            $Lower["${DMU}I"][$Factor] = $DMUData[$DMU][$Factor] ;

```



```

        $Upper["${DMU}I"][$Factor]=$DMUData[$DMU][$Factor] ;
    }
}

foreach ( array_keys( $Lower ) as $virtualDMU ) {
    if ( $iOrientation == OUTPUT_ORIENTATION ) {
        $LowerResult = CCR( $virtualDMU , 0 ,
            $Inputs , $Outputs ,
            array_keys($Lower), $iRTS ,
            0 , $Lower );

        //
        // if it is not efficient it is not on the SOA
        //
        if ( $LowerResult["Theta"] >= ( 1 + CLOSE_ENOUGH ) ) {
            unset( $Lower[$virtualDMU] );
        }
    } elseif ( $iOrientation == INPUT_ORIENTATION ) {
        $LowerResult = IO( $virtualDMU , 0 ,
            $Inputs , $Outputs ,
            array_keys($Lower) , $iRTS ,
            0 , $Lower );

        //
        // if it is not efficient it is not on the SOA
        //
        if ( $LowerResult["Theta"] <= ( 1 - CLOSE_ENOUGH ) ) {
            unset( $Lower[$virtualDMU] );
        }
    }
}

foreach ( array_keys( $Upper ) as $virtualDMU ) {
    if ( $iOrientation == OUTPUT_ORIENTATION ) {
        $UpperResult = CCR( $virtualDMU , 0 ,
            $Inputs , $Outputs ,
            array_keys($Upper), $iRTS ,
            0 , $Upper );

        //
        // if it is not efficient it is not on the SOA
        //
        if ( $UpperResult["Theta"] >= ( 1 + CLOSE_ENOUGH ) ) {
            unset( $Upper[$virtualDMU] );
        }
    } elseif ( $iOrientation == INPUT_ORIENTATION ) {
        $UpperResult = IO( $virtualDMU , 0 ,
            $Inputs , $Outputs ,
            array_keys($Upper) , $iRTS ,
            0 , $Upper );

        //
        // if it is not efficient it is not on the SOA
        //
        if ( $UpperResult["Theta"] <= ( 1 - CLOSE_ENOUGH ) ) {
            unset( $Upper[$virtualDMU] );
        }
    }
}

$Frontier['Lower'] = $Lower ;
$Frontier['Upper'] = $Upper ;
return($Frontier) ;
}

//
// This function finds the Y value based on an X value from the arrays.
//
function findY ( $xValue, $xArray, $yArray ) {

    $lowKey = 0;
    $lastKey = 0;

```

```

$highKey    = 0;
$yValue     = -1;

foreach ( array_keys($xArray) as $arrayKey ) {
    //
    // if an xValue matches something in the array, then we give it the
    // corresponding yValue, because there should not be any "repeat"
    // values in this array (this is a supposition, which should remain
    // true).
    //
    if ( $xArray[$arrayKey] == $xValue ) {
        logTrace($functionName, 101, "findY: Is a Y value for the X");
        $yValue = $yArray[$arrayKey] ;
        break ;
    } elseif ( $xArray[$arrayKey] > $xValue ) {
        $lowKey = $lastKey ;
        $highKey = $arrayKey ;
        logTrace($functionName, 102, "findY: Value $xValue is above "
            . $xArray[$lowKey] . " and below " . $xArray[$highKey] );
        break ;
    } else {
        $lastKey = $arrayKey ;
    }
}
if ( $yValue == -1 ) {
    $yRise = $yArray[$highKey] - $yArray[$lowKey] ;
    logTrace($functionName, 103, "findY: yRise - " . $yArray[$highKey] .
        " - " . $yArray[$lowKey] . " = " . $yRise);
    $xRun = $xArray[$highKey] - $xArray[$lowKey] ;
    logTrace($functionName, 104, "findY: xRun - " . $xArray[$highKey] . " - "
        . $xArray[$lowKey] . " = " . $xRun);
    $slope = $yRise / $xRun ;
    logTrace($functionName, 105, "findY: slope - $slope");
    $yValue = $yArray[$lowKey] + ( $xValue - $xArray[$lowKey] ) * $slope ;
    logTrace($functionName, 106, "findY: yValue - " . $yArray[$lowKey] .
        " + ( " . $xValue . " - " . $xArray[$lowKey] . " ) * " .
        $slope . " = " . $yValue);
}
logTrace($functionName, 108, "findY: yValue : $yValue");
return ( $yValue ) ;
}

?>

```

DataFunctions.php

```

<?php
//
// Lane Inman
// Engineering Management Department
// Portland State University
//
// $Id: DataFunctions.php,v 1.22 2004/07/04 20:35:33 oli Exp $
//

define("DAY_ANNUALIZED"      , 7); // Time units to day, but reports are annual.
define("PERIOD_AS_QUARTER"   , 6); // Time units to period (no conversion)
define("YEAR_TIME_UNITS"     , 5); // Time units to year
define("QUARTER_TIME_UNITS"  , 4); // Time units to quarter
define("MONTH_TIME_UNITS"    , 3); // Time units to month
define("WEEK_TIME_UNITS"     , 2); // Time units to week
define("DAY_TIME_UNITS"      , 1); // Time units to day
define("PERIOD_TIME_UNITS"   , 0); // Time units to period (no conversion)

//
// Define the start year for studies

```

```

//
define("FIRST_YEAR"      , 1980);

//
// Time to date takes the time in days and converts to a date...
//
function timeToDate( $Time, $iTimeUnits, $table, $connection ) {

    if ( $iTimeUnits == PERIOD_TIME_UNITS ) {
        $returnValue = $Time ;      // Nothing to change here.
    } elseif ( $iTimeUnits == PERIOD_AS_QUARTER ) {
        $Quarter      = $Time % 4 ;
        $Year          = FIRST_YEAR + ( $Time - $Quarter ) / 4 ;
        $Quarter      = $Quarter + 1 ;
        $returnValue = $Year . "Q" . $Quarter ;
        return ($returnValue);
    } elseif ( $iTimeUnits == DAY_TIME_UNITS ) {
        $queryString = "select DATE_FORMAT(Time,'%Y/%m/%d') as Time from $table
                        where TO_DAYS(Time)=$Time GROUP BY `Time`";

        $queryResult = mysql_query( $queryString, $connection) or
            die("Invalid Query:<br>\n" .
                mysql_error() . "<br>\n");

        $Row = mysql_fetch_assoc($queryResult) ;
        $returnValue = $Row["Time"] ;
    }
    return ( $returnValue ) ;
}

//
// Function gets name of DMU so we dont have to cross reference it....
//
function getName( $DMU, $Fields, $table , $connection ) {

    $queryString="select ";
    $bNoComma = 1;
    foreach ( $Fields as $Colum ) {
        if ( ! $NoComma ) {
            $queryString .= ", ";
        } else {
            $bNoComma = 0 ;
        }
        $queryString.= "$Colum" ;
    }
    $queryString .= " AS DMUNAME from $table";
    $queryResult = mysql_query( $queryString , $connection) or
        die("Invalid Query:<br>\n" .
            mysql_error() . "<br>\n");

    $Row = mysql_fetch_assoc($queryResult) ;
    return($Row[DMUNAME]) ;
}

//
// getData
//
// serverName : Database server to connect to
// userName   : User to connect to
// userPass   : Password for user
// dbName     : database name
// tableName  : table to connect
// timeUnits  : Units to do things
// inputs     : array of input identifiers
// outputs    : array of output identifiers
// time       : Time array
//
function getData ( $connection, $tableName, $timeUnits,

```

```

        $inputs      , $outputs  , $startTime,
        $endTime    , $timeColum, $whereStatement      ) {

//
// Determine the minimum time from which we will select 0
//
if ( $timeUnits == DAY_TIME_UNITS ) {
    $timeString="TO_DAYS($timeColum)";
} elseif ( $timeUnits == WEEK_TIME_UNITS ) {
    $timeString = "(YEAR($timeColum)*52 + WEEK($timeColum))";
} elseif ( $timeUnits == MONTH_TIME_UNITS ) {
    $timeString = "(YEAR($timeColum)*12 + MONTH($timeColum))";
} elseif ( $timeUnits == QUARTER_TIME_UNITS ) {
    $timeString = "( YEAR($timeColum) - " . FIRST_YEAR .
        " ) * 4 + Quarter($timeColum))";
} elseif ( $timeUnits == YEAR_TIME_UNITS ) {
    $timeString = "(YEAR($timeColum))";
} elseif ( $timeUnits == PERIOD_TIME_UNITS ) {
    $timeString = "$timeColum";
}

$queryString = "select min($timeString) from $tableName";

$queryResult = mysql_query( $queryString , $connection) or
    die("Invalid Query:<br>\n" . mysql_error() . "<br>\n");

$minTime=$Row[0];

//
// Get DMU Data
//
$fieldsToQuery=array_merge($outputs, $inputs);
$queryString="select DMU";
foreach ( $fieldsToQuery as $field ) {
    $queryString.=",$field";
}
$queryString .= ",$timeString as Time";
$queryString .= " from $tableName $whereStatement order by Time " ;
$queryResult = mysql_query("$queryString", $connection) or
    die("Invalid Query:<br>" . mysql_error() . "<br>\n");

while ( $Row = mysql_fetch_assoc($queryResult) ) {
    if ( $startTime <= $Row["Time"] && $Row["Time"] <= $endTime ) {
        $DMU_Data[$Row["DMU"]]["DMU"]=$Row["DMU"] ;
        foreach ( $fieldsToQuery as $field ) {
            $DMU_Data[$Row["DMU"]][$field]=$Row[$field] ;
        }
        $DMU_Data[$Row["DMU"]]["Time"]=$Row["Time"];
    }
}
return ( $DMU_Data ) ;
}

function getData2 ( $connection, $tableName, $timeUnits, $inputs      ,
    $outputs      , $inputsAs , $outputsAs, $timeColum ,
    $whereStatement      ) {

    $DMU_Data = array();
    $functionName="getData2";
    logTrace($functionName, 300200, "Entering...");
    logTrace($functionName, 300201, " tableName [" . $tableName
        . " ] timeUnits [" . $timeUnits . " ] timeColum [" . $timeColum . " ]");

//
// Determine the minimum time from which we will select 0
//
if ( $timeUnits == DAY_TIME_UNITS ) {

```

```

        $timeString="TO_DAYS($timeColum)";
    } elseif ( $timeUnits == WEEK_TIME_UNITS ) {
        $timeString = "(YEAR($timeColum)*52 + WEEK($timeColum))";
    } elseif ( $timeUnits == MONTH_TIME_UNITS ) {
        $timeString = "(YEAR($timeColum)*12 + MONTH($timeColum))";
    } elseif ( $timeUnits == QUARTER_TIME_UNITS ) {
        $timeString = "(YEAR($timeColum)*4 + Quarter($timeColum))";
    } elseif ( $timeUnits == YEAR_TIME_UNITS ) {
        $timeString = "(YEAR($timeColum))";
    } elseif ( $timeUnits == PERIOD_TIME_UNITS ) {
        $timeString = "$timeColum";
    }
}

logTrace($functionName, 300210, " timeString [" . $timeString . "]);

//
// Get DMU Data
//
$fieldsToQuery=array_merge($outputs , $inputs);
$fieldEquivs = array_merge($outputsAs, $inputsAs );
$queryString="select DMU";
foreach (array_keys($fieldsToQuery) as $field ) {
    if ( $fieldEquivs[$field] != "" ) {
        $queryString .= "," . $fieldEquivs[$field] . " as "
            . $fieldsToQuery[$field] ;
    } else {
        $queryString.="," . $fieldsToQuery[$field];
    }
}

$queryString .= ",".$timeString as Time";
$queryString .= " from $tableName $whereStatement order by Time " ;

logTrace($functionName, 300211, " queryString [" . $queryString . " ] ");

$queryResult = mysql_query("$queryString", $connection) or
    die("Invalid Query:<br>" . mysql_error() . "<br>\n");

while ( $Row = mysql_fetch_assoc($queryResult) ) {
    $DMU_Data[$Row["DMU"]]["DMU"]=$Row["DMU"] ;
    foreach ( $fieldsToQuery as $field ) {
        $DMU_Data[$Row["DMU"]][$field]=$Row[$field] ;
    }
    $DMU_Data[$Row["DMU"]]["Time"]=$Row["Time"];
}

logTrace($functionName, 300299, "Exiting...");
return ($DMU_Data) ;
}

function getUnixTimeStamp ( $DateString , $connection ) {

    $queryString = "select UNIX_TIMESTAMP('$DateString 23:59:59') as TIME";
    $queryResult = mysql_query("$queryString", $connection ) or
        die ("Invalid Query:<br>" . mysql_error() . "<br>\n");

    $Row = mysql_fetch_assoc($queryResult);
    return($Row[TIME]);
}

function getDateFromDay ( $dayNumber, $connection ) {
    // echo "Day Number : $dayNumber \n";
    $queryString = "select FROM_DAYS($dayNumber) as date";
    $queryResult = mysql_query("$queryString", $connection ) or
        die("Invalid Query:<br>" . mysql_error() . "<br>\n");

    $Row = mysql_fetch_assoc($queryResult);

```

```

    return($Row["date"]);
}

function getDayFromDate ( $dateString , $connection ) {

    $functionName = "getDayFromDate";
    logTrace($functionName, 300100, "Entering");
    logTrace($functionName, 300101, " dateString [" . $dateString . ] " );

    $queryString = "select TO_DAYS('$dateString 23:59:59') as DAY";

    logTrace( $functionName, 300102, " queryString [" . $queryString . ] " );

    $queryResult = mysql_query("$queryString", $connection ) or
        die ("Invalid Query:<br>" . mysql_error() . "<br>\n");

    $Row = mysql_fetch_assoc($queryResult);
    logTrace( $functionName, 300103, " Row[DAY] [" . $Row['DAY'] . ] " );

    logTrace( $functionName, 300199, "Exiting...");
    return($Row["DAY"]);
}

function getPeriod( $stamp          , $connection ) {
    if ( $_SESSION["TimeUnits"] == DAY_TIME_UNITS ) {
        $returnValue = getDayFromDate( $stamp, $connection);
    } elseif ( $_SESSION["TimeUnits"] == PERIOD_AS_QUARTER ) {
        $year          = substr($stamp, 0, strpos($stamp, "Q")      );
        $quar          = substr($stamp, ( strpos($stamp, "Q") + 1 ) );
        $quar          -= 1 ;
        $returnValue = ($year - FIRST_YEAR) * 4 + $quar      ;
    } else {
        $returnValue = $stamp ;
    }
    return($returnValue);
}

function getStamp ( $periodNumber, $connection ) {

    $functionName="getStamp";
    logTrace($functionName, 300400, "Entering...");
    logTrace($functionName, 300401, " periodNumber [" . $periodNumber . ]");
    if ( $_SESSION["TimeUnits"] == DAY_TIME_UNITS ) {

        $returnValue = ( $periodNumber > 9999999999 ) ? "INF" :
            getDateFromDay( $periodNumber, $connection ) ;
    } elseif ( $_SESSION["TimeUnits"] == PERIOD_AS_QUARTER ) {
        $period = $periodNumber % 4 ;
        $year   = ( $periodNumber - $period ) / 4 ;
        $period += 1;
        $returnValue = FIRST_YEAR + $year . "Q" . $period ;
    } elseif ( $_SESSION["TimeUnits"] == PERIOD_TIME_UNITS ) {
        $returnValue = $periodNumber ;
    } elseif ( $_SESSION["TimeUnits"] == YEAR_TIME_UNITS ) {
        $returnValue = $periodNumber ;
    }
    logTrace($functionName, 300498, " returnValue [" . $returnValue . ]");
    logTrace($functionName, 300499, "Exitting...");

    return $returnValue ;
}

function getField ( $fieldToMatch, $matchValue, $fieldToGet, $tableName,
    $connection ) {

    $functionName = "getField" ;
    logTrace($functionName, 300000, "Entering");

```

```

logTrace($functionName, 300001, " fieldToMatch [" . $fieldToMatch ."] " .
      " matchValue [" . $matchValue . " ] " .
      " fieldToGet [" . $fieldToGet . " ] " );

if ( preg_match('/[A-Za-z]/',$matchValue) ) {
    $queryString = "select $fieldToGet from $tableName" .
        " where ` $fieldToMatch `=' $matchValue '";
} else {
    $queryString = "select $fieldToGet from $tableName"
        . " where $fieldToMatch=$matchValue";
}
logTrace($functionName, 300002, " queryString [" . $queryString . "]);

$queryResult = mysql_query("$queryString", $connection ) or
    die ("Invalid Query:<br>" . mysql_error() . "<br>\n");

$row = mysql_fetch_assoc($queryResult);
logTrace($functionName, 300003, "Returning [" . $row[$fieldToGet] . "]);
logTrace($functionName, 300004, "Exiting...");
return($row[$fieldToGet]);
}

?>

```

LogFunctions.php

```

<?php
// $Id: LogFunctions.php,v 1.9 2004/06/15 19:38:30 oli Exp $
//

define("LOG_NORM" , 0 );
define("LOG_TRACE", 1 );

function logTrace ( $functionName, $messageID, $message ) {
    //
    // Added isset to make sure that there is something set, if there is not
    // a specific value set then we will assume normal logging and thus not
    // report anything.
    //
    $functionLogLevel = isset($_SESSION[$functionName]) ?
        $_SESSION["$functionName"] : LOG_NORM ;
    if ( $_SESSION["logLevel"] == LOG_TRACE ||
        $functionLogLevel == LOG_TRACE ) {
        logMessage( $functionName, $messageID, $message ) ;
    }
}

function logMessage ( $functionName, $messageID, $message ) {
    echo $_SESSION["logLevel"] . ":" . $functionName . ":" . $messageID
        . ":" . $message ;
    if ( $_SESSION["displayFormat"] == DISPLAY_TEXT ) {
        echo "\n";
    } elseif ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "<br>\n";
    }
}

function reportMessage ( $message ) {
    echo $message ;
    if ( $_SESSION["displayFormat"] == DISPLAY_TEXT ) {
        echo "\n";
    } elseif ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "<br>\n";
    }
}

```

```

function tableHeader ( $title, $label, $array ) {
    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "<table cellpadding=0 cellspacing=0>\n"
        . "    <tr bgcolor=\"black\">\n"
        . "        <td colspan=" . (count($array) + 1)
        . " style=\"color:white\">\n"
        . "            <font style=\"color:white\">\n"
        . "                " . $title . "\n"
        . "            </font>\n"
        . "        </td>\n"
        . "    </tr>\n";
        echo "    <tr bgcolor=\"black\" >\n"
        . "        <td>\n"
        . "            <font style=\"color:white\">\n"
        . "                <b>" . $label . "</b>\n"
        . "            </font>\n"
        . "        </td>\n";

        foreach ( $array as $element ) {
            echo "            <td>\n"
            . "                <font style=\"color:white\">\n"
            . "                    <b>" . $element . "</b>\n"
            . "                </font>\n"
            . "            </td>\n";
        }
        echo "\n";
    } else {
        for ( $x = 1 ; $x <= strlen($title) ; $x++ ) {
            echo "=";
        }
        echo "\n";
        echo $title . "\n";
        for ( $x = 1 ; $x <= strlen($title) ; $x++ ) {
            echo "=";
        }
        echo "\n";
        echo $label ;
        foreach ( $array as $element ) {
            echo ":" . $element;
        }
        echo "\n";
    }
}

function tableFooter ( $footer ) {
    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "</table>\n";
    }
}

function tableRow ( $title, $array, $colorNumber, $highlight ) {

    $colorWheel = array( "white", "silver" );
    $colorCount = count($colorWheel) ;

    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "    <tr bgcolor=\"" . $colorWheel[( $colorNumber % $colorCount)]
        . "\">\n";
        echo "        <td>" . $title . "</td>\n";
    } else {
        echo $title . ":";
    }
}

foreach ( $array as $element ) {
    if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {
        echo "        <td align=center>" . $element . "</td>\n";
    } else {
        echo $element . ":" ;
    }
}

```



```
    }  
  }  
  if ( $_SESSION["displayFormat"] == DISPLAY_HTML ) {  
    echo "    </tr>\n";  
  } else {  
    echo "\n";  
  }  
}  
?>
```

This is a place holder.

[1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20>(* new but
Not)[21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][39][40][38][41]
[42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62]
[63][64][66][67][68][69][70][71][73][72][74][75][76>(* new but not)][77][78][79][80]
[81][82][87][83][84][85][86][88>(* new but) [89][90][91][92][93][94][95][96][97][98]
[99][100]*new but [101]*[102][103][104][105]*[106][107][108][109][110][111][112]
[113][114][115]