

# Inverse and Stochastic DEA: Models and Applications

**A Thesis**

submitted to

Indian Institute of Science Education and Research Pune

in partial fulfillment of the requirements for the

MS(with Dissertation) Programme

by

**Bittu Kumar**

20202027



Indian Institute of Science Education and Research Pune

Dr. Homi Bhabha Road,  
Pashan, Pune 411008, INDIA.

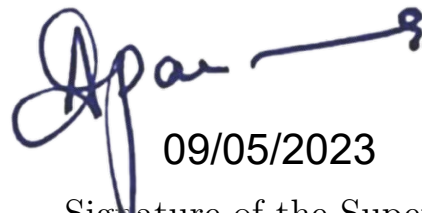
May, 2023

Supervisor: **Dr. Aparna Mehra**  
Department of Mathematics, IIT Delhi

© Bittu Kumar 2023  
All rights reserved

# Certificate

This is to certify that this dissertation entitled **Inverse and Stochastic DEA: Models and Applications** towards the partial fulfilment of the MS(by Dissertation) degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Bittu Kumar at Indian Institute of Technology, Delhi under the supervision of Dr. Aparna Mehra, Associate Professor, Department of Mathematics, during the academic year 2022-2023.

A handwritten signature in blue ink, appearing to read 'Aparna', with a long horizontal stroke extending to the right.

09/05/2023

Signature of the Supervisor

## Committee:

Supervisor: Dr. Aparna Mehra

Local Guide: Dr. Soumen Maity

This thesis is dedicated to  
My Family

# Declaration

I hereby declare that the matter embodied in the report entitled **Inverse and Stochastic DEA: Models and Applications** are the results of the work carried out by me at the Department of Mathematics, Indian Institute of Technology, Delhi, under the supervision of **Dr. Aparna Mehra** and the same has not been submitted elsewhere for any other degree.

*Bittu Kumar*

**Bittu Kumar**  
Date: 09.05.2023

# Abstract

This thesis offers an in-depth investigation of Data Envelopment Analysis (DEA) models, encompassing classical DEA, inverse DEA, and stochastic DEA. The objective is to develop and apply inverse DEA models in the context of stochastic data. The research commences with a comprehensive introduction to classical DEA models, such as CCR, BCC, Additive, and SBM models. Subsequently, the study delves into inverse DEA and its application through a case study involving 15 hypothetical retail stores.

Further, the thesis presents stochastic DEA models that account for data uncertainty and illustrates their practicality in real-world situations, as demonstrated by a case involving 20 bank branches. The primary contribution of this research lies in the analysis of the 15 hypothetical stores for budgeting and planning purposes, as well as the development of novel inverse DEA models that integrate stochastic data. This approach expands the applicability of DEA models in uncertain environments, assuming a symmetric error structure.

For practical application and replication purposes, Python code implementations for each DEA model are provided in the Appendix. The proposed models hold potential for further research within dynamic and network DEA frameworks and have practical applications across various industries.

# Contents

0.1 Notations	6
<b>1 Introduction</b>	<b>1</b>
1.1 Background	1
1.2 Classical DEA	2
1.3 Inverse DEA	2
1.4 Stochastic DEA	2
1.5 Research Objective and contributions	3
1.6 Thesis structure	3
1.7 Conclusions	4
<b>2 Classical DEA Models, Formulation, and Key Terms</b>	<b>5</b>
2.1 Basic DEA Models	9
2.1.1 CCR Models	9
2.1.2 BCC models	14
2.1.3 Additive Models	16
2.1.4 SBM Models	17
2.2 Conclusions	19

<b>3 Inverse DEA</b>	<b>21</b>
3.1 Classical Inverse DEA Models . . . . .	22
3.2 Inverse DEA for Budgeting and Planning . . . . .	28
3.2.1 Income-based Inverse DEA Model . . . . .	28
3.2.2 Budget-based inverse DEA model . . . . .	31
3.3 Applicatons of Income-based and Budget-based Models . . . . .	33
<b>4 Stochastic DEA</b>	<b>39</b>
4.1 Developing an input-oriented DEA model [11] . . . . .	40
4.2 Linearization of Stochastic DEA through Symmetric Error Structure (3) . . . . .	45
<b>5 Inverse DEA with Stochastic data</b>	<b>55</b>
5.1 Inverse DEA in the presence of stochastic data . . . . .	57
5.1.1 Input level estimation . . . . .	58
5.1.2 Output level estimation . . . . .	62
5.2 Application in Banking Sector . . . . .	66
<b>6 Conclusion and Future Research</b>	<b>72</b>
6.1 Summary of Contributions: . . . . .	72
6.2 Future Research . . . . .	72
<b>A Code listings</b>	<b>76</b>

<b>A.1 Python Codes for CCR, BCC, ADDITIVE, and SBM models</b> . . . . .	76
<b>A.1.1 CCR Codes</b> . . . . .	76
<b>A.1.2 BCC Codes</b> . . . . .	78
<b>A.1.3 Additive model Code</b> . . . . .	81
<b>A.1.4 SBM codes</b> . . . . .	82
<b>A.2 Python Codes for inverse DEA for data of 15 Stores</b> . . . . .	84
<b>A.3 university A efficiency calculation code</b> . . . . .	88
<b>A.4 Efficiency calculation Code for 20 DMUs</b> . . . . .	89
<b>A.5 Code for input level estimation in case of Stochastic data</b> . . . . .	92
<b>B Data</b>	<b>94</b>
<b>B.1 Hypothetical data for the three universities</b> . . . . .	94
<b>B.2 Data of 20 Banks</b> . . . . .	94
<b>C Models</b>	<b>97</b>



# List of Tables

2.1	BCC Models [8]	15
2.2	Additive model and Dual problem	17
3.1	Generalized Input-oriented and output-oriented DEA models	22
3.2	Store Data with Input and Output Efficiencies	34
3.3	Results after applying income-based model with A=80,000	35
3.4	Analysis of Store 13 for Different Income Levels	36
3.5	Results of the budget-based model with B=80,000	37
3.6	Store 13 under different budget levels	37
4.1	Optimal solution table	44
4.2	input-oriented and output-oriented CCR radial-form efficiency models	46
4.3	Vectors, random vectors, expected values, and distributions	46
4.4	Input and output-oriented chance constraint CCR SDEA	46
4.5	Stochastic efficiency linear models	52
4.6	Efficiency results for each DMU with different significance levels	54
5.1	LP and MLOP form of Chance-constrained Inverse DEA in significance level $\alpha$	65

5.2	The percentage of expected increases in the output levels for DMU10.	67
5.3	The percentage of necessary increases in the inputs for DMU10. . . .	68
5.4	The percentage of expected increase in the outputs for DMU18. . . .	69
5.5	The percentage of necessary increases in the inputs for DMU18. . . .	69
B.1	Hypothetical data for the input and outputs for the three universities	
	[11] . . . . .	94
B.2	Hypothetical data for the stochastic input (budget) for the three uni-	
	versities [11] . . . . .	94
B.3	The inputs of 20 bank branches. [13] . . . . .	95
B.4	The outputs of 20 bank branches. [12] . . . . .	96
C.1	Generalized Input and Output Oriented inverse DEA Models . . . .	97
C.2	Stochastic Input-Output oriented Models . . . . .	97

## 0.1 Notations

**DMU<sub>j</sub>**:  $j = 1, 2, \dots, n$  Number of Decision-Making units

**PPS** Production Possibility Set (Technology Set)

**DMU<sub>j</sub>** the  $j^{\text{th}}$  DMU

**DMU<sub>o</sub>** the DMU under consideration (Target DMU) for evaluation

**X** =  $(x_{ij}) \in R^{m \times n}$  Input vector matrix

**Y** =  $(y_{ij}) \in R^{s \times n}$  Output vector matrix

**x<sub>j</sub>** =  $(x_{1j}, \dots, x_{mj})$  the input vector of DMU<sub>j</sub>

**x<sub>o</sub>** =  $(x_{1o}, \dots, x_{mo})$  the input vector of target DMU<sub>o</sub>

**y<sub>j</sub>** =  $(y_{1j}, \dots, y_{sj})$  the output vector of DMU<sub>j</sub>

**y<sub>o</sub>** =  $(y_{1o}, \dots, y_{so})$  the output vector of target DMU<sub>o</sub>

**v** =  $(u_1, \dots, u_s)$  the vector of output weights

**u** =  $(v_1, \dots, v_m)$  the vector of input weights

**CRS** Constant Return-to-scale

**VRS** Variable Return-to-Scale

**NIRS** Non-Increasing Return-to-Scale

**NDRS** Non-Decreasing Return-to-Scale

**LP** Linear Programming

**CCR** Charnes, Cooper, and Rhodes

**BCC** Banker, Charnes, and Cooper

**SBM** Slacks-based measures

# Acknowledgements

I would like to express my deepest gratitude to my guide, Dr. Aparna Mehra, whose expertise, understanding, and guidance have made this thesis possible. Her encouragement and constant feedback were instrumental in the successful completion of this work.

I would also like to extend my heartfelt appreciation to my local guide, Dr. Soumen Maity. The courses he taught me and the previous project I undertook under his supervision significantly contributed to my ability to face the challenges in this project. His valuable insights, unwavering support, and dedication to my academic growth have been truly inspiring and played a vital role in the completion of my thesis.

My sincere thanks go to my family, who have always been there for me and supported me unconditionally. Their love and encouragement have been a constant source of strength and motivation.

I am thankful to my classmates and seniors, particularly Tushar Arora, who was always ready to help me. I also want to express my gratitude to my friends at IIT Delhi for their assistance during my stay there. Lastly, I would like to thank Pratiksha Singh from IIT-ISM Dhanbad for her help in organizing my thesis and correcting grammatical errors.

# Chapter 1

## Introduction

### 1.1 Background

Data Envelopment Analysis (DEA) has emerged as a powerful tool for evaluating the relative efficiency of decision-making units (DMUs) in various sectors such as education, healthcare, banking, and transportation. The fundamental principle of efficiency is to achieve the best possible outcome with the least utilization of resources. Efficiency measurement and enhancement is, therefore, crucial for making better and more accurate decisions for any decision-making unit. However, quantifying efficiency is a complex task, particularly when dealing with multiple inputs and outputs [15].

About fifty years ago, Farrel introduced the assessment of productivity efficiency in his seminal paper “The measurement of productive efficiency”. Two decades later, in 1978, Charnes et al. [4] presented Data Envelopment Analysis (DEA), a methodology for computing the relative efficiency of units with multiple outputs and inputs. DEA defines efficiency as the ratio of the weighted sum of outputs to the weighted sum of inputs and provides a mathematical representation in the forthcoming Chapters. Some notable advantages of DEA include [15]:

1. The ability to handle multiple outputs and inputs with different units.
2. The capability to work with both qualitative and quantitative data.
3. Serving as an effective decision-making tool for management by highlighting areas for improvement.

In light of these advantages, DEA has garnered significant attention from scientists and researchers in the fields of decision-making and operations research. DEA is a

non-parametric linear programming method that evaluates the efficiency of DMUs by comparing their input-output ratios. However, traditional DEA models assume deterministic inputs and outputs, which might not accurately capture the real-world uncertainties and stochastic nature of data. This thesis aims to delve deeper into DEA and its various extensions, including classical DEA models, inverse DEA, and stochastic DEA, while exploring their applications in diverse contexts. By doing so, we strive to contribute to a better understanding of efficiency measurement and provide valuable tools for organizations to optimize their resources and enhance their performance.

## **1.2 Classical DEA**

Classical DEA models, such as the CCR and BCC models, have been widely used to assess the relative efficiency of DMUs. These models are based on the assumption that inputs and outputs are deterministic and do not consider any uncertainty in the data. Despite their widespread application, classical DEA models have limitations when dealing with uncertain or stochastic data.

## **1.3 Inverse DEA**

Inverse DEA is an extension of the classical DEA, which focuses on estimating the required changes in input or output levels to achieve a desired efficiency level. This approach is particularly useful for budgeting, resource allocation, and planning purposes.

## **1.4 Stochastic DEA**

Stochastic DEA is an advancement of the classical DEA model that takes into account the stochastic nature of input and output data. By incorporating random errors or uncertainties in the data, stochastic DEA provides a more realistic assessment of

DMU efficiency.

## 1.5 Research Objective and contributions

The primary objective of this research is to develop and apply inverse DEA models in the presence of stochastic data . The main contributions of this thesis include the development of basic DEA models With python Python code attached in Appendix A (Code Listings) ,This Thesis explore and apply inverse DEA models in the context of hypothetical data for 15 stores with two inputs and two outputs, analyzing the work of Ghomi et al. [13](#) on inverse DEA in the presence of stochastic data, and providing Python code for the relevant models in the appendix section. This research contributes to the practical application of inverse DEA models in various industries, which can be used efficiently in Resourse allocation and Forecasting.

## 1.6 Thesis structure

The thesis is structured as follows:

1. Introduction: Provides an overview of DEA and its various extensions, setting the stage for the development and application of inverse DEA models in the presence of stochastic data.
2. Classical DEA Models : A comprehensive introduction to classical DEA models, including the CCR and BCC models, along with Python Code in The Appendix section .
3. Inverse DEA and Application in Stores: Presents the concept of inverse DEA and its application to efficiency analysis in a case study of 15 hypothetical Stores Data.
4. Stochastic DEA and Examples: Introduces stochastic DEA models that account for the uncertainty in data, and provides examples to demonstrate their utility in real-world scenarios.

5. Combining Inverse DEA and Stochastic DEA: Develops new inverse DEA models that incorporate stochastic data, thereby enhancing the applicability of DEA models in uncertain environments.
6. Conclusion: Highlights the key contributions and conclusions of the thesis, examines the consequences of the created models, and gives possible directions for further investigation.

## 1.7 Conclusions

The introductory chapter provides an overview of the DEA and its various extensions, setting the stage for the development and application of inverse DEA models in the presence of stochastic data. The subsequent chapters delve into the details of classical DEA, inverse DEA, and stochastic DEA, culminating in the proposed inverse DEA models with stochastic data and their real-world applications.



# Chapter 2

## Classical DEA Models, Formulation, and Key Terms

This chapter focuses on introducing the Classical DEA models and their key terms. DEA models have been widely used to measure the relative efficiency of Decision-Making Units (DMUs) in various fields, including healthcare, education, and finance. The motivation behind developing DEA models was to find a non-parametric approach to evaluate DMUs' efficiency and provide insights into improving their performance.

To understand the DEA models, key terms need to be defined, such as input and output vectors, production possibility set CRS (constant returns to scale), VRS (variable returns to scale), and technical efficiency. This chapter will explore these concepts and their significance in DEA models.

The chapter will then delve into the first classical DEA model, CCR, which assumes CRS and constant input/output prices. Then, will introduce the BCC model, which allows for VRS and varying input/output prices. We will also discuss additive models, SBM models, and Russell's measure. Each model will be explained with their benefits and shortcomings.

Overall, this chapter aims to provide us with a comprehensive understanding of DEA models, their key terms, and the different types of models that have been developed. Understanding DEA models and their key concepts can provide insights into the relative efficiency of DMUs and assist decision makers in making informed decisions to enhance their performance.

Before introducing the basic models of DEA, we need to define some key terms that are essential for understanding the concept and methodology of efficiency and productivity measurement. These terms are DMU, input, output, benchmarking, efficiency and productivity, production possibility set, returns to scale, Units, and Translation Invariant models, Radial and Non-Radial models, etc.

A DMU is an entity that uses multiple inputs to produce multiple outputs under its control. A DMU can be an individual, a group, an organization, or any unit that performs some activities or tasks. For example, a hospital can be considered a DMU that uses inputs such as doctors, nurses, beds, equipment, etc., to produce outputs such as patients treated, surgeries performed, quality indicators, etc. (Cooper et al., 2007).

Input is any resource that is used by a DMU to produce outputs. An input can be measured in physical units (such as labor hours) or monetary units (such as cost). Input can also be classified into variable inputs (such as materials) or fixed inputs (such as capital). The choice and measurement of inputs depend on the objective and context of the analysis (Zhu, 2014).

Output is any product or service that is produced by a DMU using inputs. Output can also be measured in physical units (such as quantity) or monetary units (such as revenue). Output can also be classified into desirable outputs (such as profits) or undesirable outputs (such as emissions). The choice and measurement of outputs also depend on the objective and context of the analysis (Cooper et al., 2007).

Benchmarking is a process of comparing the performance of a DMU with that of other DMUs or with some standards or best practices. Benchmarking can help identify the strengths and weaknesses of a DMU and provide insights for improvement. Benchmarking can also be used for ranking, rating, or classifying DMUs based on their efficiency scores or other indicators (Zhu, 2014).

Efficiency and productivity are two related measures of how well a DMU uses its inputs to produce outputs. Efficiency can be expressed as the ratio of actual output to potential output or as the ratio of minimum input to actual input. Productivity can be expressed as the ratio of output to input or as the ratio of output growth

to input growth. Efficiency and productivity can also be decomposed into technical efficiency and allocative efficiency for efficiency; and efficiency change and technical change for productivity. Technical efficiency reflects the ability of a DMU to produce the maximum output from a given input or to use the minimum input for a given output. Allocative efficiency reflects the ability of a DMU to choose the optimal mix of inputs and outputs given their prices. The technical change reflects the shift or movement of the best practice frontier due to innovation or technological progress. Efficiency change reflects the improvement or deterioration of a DMU's performance relative to the best practice frontier (Cooper et al., 2007).

A production possibility set is a set of all feasible combinations of inputs and outputs for a given technology. A production possibility set defines what can be produced by a DMU given its available resources and constraints. A production possibility set can have different shapes depending on the assumptions about returns to scale, convexity, and disposability (Zhu,2014). Returns to scale refer to how output changes when inputs are proportionally increased or decreased. There are four types of returns to scale: constant returns to scale (CRS), variable returns to scale (VRS), non-increasing returns to scale (NIRS), and non-decreasing returns to scale (NDRS). CRS means that output changes in the same proportion as inputs; VRS means that output changes more than or less than inputs; NIRS means that output changes less than or equal to inputs; and NDRS means that output varies more than or equal to inputs. Returns to scale affect the shape and properties of the production possibility set and have implications for efficiency measurement (Zhu, 2014).

**0.1** PPS is generally defined as **1**.

$$T = \{(x, y) \in \mathbb{R}_+^m \times \mathbb{R}_+^s : \text{the output } y \text{ can be produced by input } x\} \quad (2.1)$$

Some axiomatic assumptions that underlie various formulations of the technology set which are [referred to Banker (1984), Banker et al. (1984), and Banker and Thrall (1992)]

1. the Inclusion of Observations.
2. Convexity
3. Different return to scale Possibilities
4. Free Disposability.

## 5. Minimal extrapolation

Based on these Five axioms, a unique [0.1](#) PPS represented as :

$$T^{(p)} = \{(x, y) \in \mathbb{R}_+^m \times \mathbb{R}_+^s : \exists \lambda = (\lambda_1, \dots, \lambda_n) \in S(p) : \sum_{j=1}^n \lambda_j x_j \leq X, \sum_{j=1}^n \lambda_j y_j \geq y\} \quad (2.2)$$

[1](#)

where  $p \in \{CRS, VRS, NIRS, NDRS\}$  [0.1](#) and

$$\begin{aligned} S(CRS) &= \{\lambda = (\lambda_1, \dots, \lambda_n) : \lambda_j \geq 0, j = 1, \dots, n\}, \\ S(VRS) &= \{\lambda = (\lambda_1, \dots, \lambda_n) : \sum_{j=1}^n \lambda_j = 1, \lambda_j \geq 0, j = 1, \dots, n\}, \\ S(NIRS) &= \{\lambda = (\lambda_1, \dots, \lambda_n) : \sum_{j=1}^n \lambda_j \leq 1, \lambda_j \geq 0, j = 1, \dots, n\}, \\ S(NDRS) &= \{\lambda = (\lambda_1, \dots, \lambda_n) : \sum_{j=1}^n \lambda_j = 1, \lambda_j \in [0, 1], j = 1, \dots, n\} \end{aligned}$$

Unit invariant DEA models are those that are independent of the units in which the input and output variables are measured, while translation invariant DEA models are those that are independent of an affine translation of the input and output variables. These properties are desirable for ensuring consistency and robustness of the efficiency scores.

Radial DEA models are those that measure the relative efficiency of decision-making units (DMUs) by assuming proportional changes in inputs or outputs. They neglect the effect of slacks (input excesses or output shortfalls) in evaluating efficiency [17](#). Non-radial DEA models [21](#) are those that measure the relative efficiency of DMUs by considering the effect of slacks in evaluating efficiency. They deal directly with input excesses and output shortfalls by using different measures such as Russell Measure, Slacks-based Measure, etc.

## 2.1 Basic DEA Models

DEA models are based on the idea of comparing the performance of a set of DMUs (6) that use inputs to produce outputs. A common measure of efficiency used in DEA is the ratio of weighted outputs to weighted inputs, where the weights are determined by solving a linear programming problem. In the single input-output case, this ratio reduces to the output per unit of input. In the multiple input-output cases, this ratio reflects the trade-offs between different inputs and outputs. DEA models can be formulated in different ways depending on the orientation (input or output), the returns to scale assumption (constant, variable, or general), and the type of efficiency measure (radial or non-radial).

### 2.1.1 CCR Models

(8) The CCR model is a basic DEA model proposed by Charnes, Cooper, and Rhodes in 1978. It uses virtual inputs and outputs formed by weights to determine the optimal weights for each DMU using LP. The weights are derived from the data and may vary from one DMU to another. Considering a set of  $n$  DMUs (6) with each  $DMU_j$  using  $m$  inputs  $x_{ij}$  ( $i = 1, 2, \dots, m$ ) and producing  $s$  outputs  $y_{rj}$  ( $r = 1, 2, \dots, s$ ) and for each DMUs we construct virtual input and virtual outputs with unknown input weights  $v_i$  ( $i = 1, 2, \dots, m$ ) and unknown output weights  $u_r$  ( $r = 1, 2, \dots, s$ ) as follows:

$$\begin{aligned}\text{Virtual Input} &= \sum_i v_i x_{ij} \\ \text{Virtual Output} &= \sum_r u_r y_{rj}\end{aligned}$$

From the given data to measure the efficiency of each DMUs once, So we need to perform  $n$  optimizations. Let  $DMU_o$  ( $o \in (1, 2, \dots, n)$ ). Then we need to solve the following fractional programming problem to obtain the value of each input and

output weight.

$$\begin{aligned}
\text{(FP-CCR);maximize : } \theta &= \frac{\sum_{r=1}^s u_r y_{ro}}{\sum_{i=1}^m v_i x_{io}} \\
\text{Subject to } \frac{\sum_{r=1}^s u_r y_{rj}}{\sum_{i=1}^m v_i x_{ij}} &\leq 1 \quad (j = 1, 2, \dots, n) \\
\mathbf{u} &\geq 0 \\
\mathbf{v} &\geq 0
\end{aligned} \tag{2.3}$$

Where  $\mathbf{u}$  and  $\mathbf{v}$  are row vectors of output and input multipliers(weights). The (2.3) is equivalent to following linear program:

$$\begin{aligned}
\text{(LP-CCR);maximize : } \theta &= \sum_{r=1}^s u_r y_{ro} \\
\text{Subject to } \sum_{i=1}^m v_i x_{io} &= 1 \\
\sum_{r=1}^s u_r y_{rj} - \sum_{i=1}^m v_i x_{ij} &\leq 0 \quad (j = 1, 2, \dots, n) \\
\mathbf{u} &\geq 0 \\
\mathbf{v} &\geq 0
\end{aligned} \tag{2.4}$$

The optimal value  $\theta^*$  of the equivalent program (2.3) and (2.4) are the same.

**Theorem 1** (Unit Invariance Theorem [8]). *The optimal value  $\theta^*$  of (2.3) and (2.4) are unit invariant if all DMUs use the same units for inputs and outputs.*

Now using vector notation  $\mathbf{v}$  and  $\mathbf{u}$  input and output weights (multiplier).The linear program (2.4) can be written as follows:

$$\begin{aligned}
\text{[Multiplier form of CCR] Maximize } &\mathbf{u}\mathbf{y}_o \\
\text{subject to } &\mathbf{v}\mathbf{x}_o = 1 \\
&-\mathbf{v}\mathbf{X} + \mathbf{u}\mathbf{Y} \leq \mathbf{0} \\
&\mathbf{v} \geq 0, \mathbf{u} \geq 0
\end{aligned} \tag{2.5}$$

The dual of above (2.5) is known as the Envelopment form of CCR models as shown below

$$\begin{aligned}
& \text{[Envelopment form of CCR / Radial Form]} \quad \text{Minimize: } \theta \\
& \text{Subject to: } \theta \mathbf{x}_o - \mathbf{X}\lambda \geq \mathbf{0} \\
& \quad \mathbf{Y}\lambda - \mathbf{y}_o \geq \mathbf{0} \\
& \quad \lambda \geq \mathbf{0}
\end{aligned} \tag{2.6}$$

Where  $\lambda = (\lambda_1 \dots \lambda_n)^T$  and real number  $\theta$  are dual variables. The LPP (2.6) is feasible ( $\lambda_o = 0, \lambda_i = 0 (i \neq o), \theta = 1$ ). and hence optimal value will not exceed 1 and we observe that  $0 < \theta^* \leq 1$ . Also, we can see the relationship between  $T^{(CRS)}$  and by looking at the constraints of (2.6) we have  $(\theta \mathbf{x}_o, \mathbf{y}_o) \in T^{(CRS)}$ . The objective is to find the minimum  $\theta$  that reduces the input vector radially to  $\theta x_o$  while remaining in  $T^{CRS}$ . We are looking for activity in  $T^{CRS}$  that guarantees at least the output level  $y_o$  of  $DMU_o$  in all components while proportionally (radially) reducing the input vector  $x_o$  to a value as small as possible. Under the assumption of  $T^{(CRS)}$ , we can say that  $(\mathbf{X}\lambda, \mathbf{Y}\lambda)$  outperform  $(\theta \mathbf{x}_o, \mathbf{y}_o)$  when  $\theta^* < 1$ . In this context, we define input excess  $\mathbf{s}^- \in \mathbf{R}^m$  and output shortfall  $\mathbf{s}^+ \in \mathbf{R}^s$  as “slack” vectors by  $\mathbf{s}^- = \theta \mathbf{x}_o - \mathbf{X}\lambda \geq \mathbf{0}$  and  $\mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_o \geq \mathbf{0}$ . To identify potential input excess and output shortfall we solve a two-phase LP.

Phase I: involves solving the dual linear programming problem (2.6) and obtaining the optimal objective value,  $\theta^*$ , which represents CCR-efficiency or “Farrell Efficiency.”

Phase II: uses the value to solve an LP problem, maximizing the sum of input excesses and output shortfalls while maintaining  $\theta = \theta^*$ . [8]

$$\begin{aligned}
& \text{Maximize: } \omega = \mathbf{s}^- + \mathbf{s}^+ \\
& \text{Subject to: } \mathbf{s}^- = \theta^* \mathbf{x}_o - \mathbf{X}\lambda \\
& \quad \mathbf{s}^- = \mathbf{Y}\lambda - \mathbf{y}_o \\
& \quad \lambda \geq \mathbf{0} \quad \mathbf{s}^- \geq \mathbf{0} \quad \mathbf{s}^+ \geq \mathbf{0}
\end{aligned} \tag{2.7}$$

The objective function can be modified with a weighted sum of input excesses and output shortfalls, which may result in different optimal solutions but will still identify nonzero slacks associated with inefficiency. [8]

**Definition 1. (Max-slack Solution, Zero-slack Activity)** An optimal solution  $(\lambda^*, \mathbf{s}^-, \mathbf{s}^{*+})$  of Phase II is called the max-slack solution. If the max-slack solution satisfies  $\mathbf{s}^{*-} = \mathbf{0}$  and  $\mathbf{s}^{*+} = \mathbf{0}$ , then it is called zero-slack.

[8]

**Definition 2. (CCR-Efficiency, Radial Efficiency, Technical Efficiency)**

A DMU<sub>o</sub> is considered CCR-efficient if the optimal solution  $(\theta^*, \lambda^*, s^{*-}, s^{*+})$  of the two LPs satisfies  $\theta^* = 1$  and is zero-slack ( $s^{*-} = 0, s^{*+} = 0$ ). Otherwise, DMU<sub>o</sub> is called CCR-inefficient because both (i)  $\theta^* = 1$  and (ii) all slacks are zero must be satisfied for full efficiency. The first condition, “radial efficiency,” is also called “technical efficiency” as a value of  $\theta^* < 1$  implies that all inputs can be reduced simultaneously without changing their proportions. Any further reductions related to nonzero slacks would change input proportions, leading to “mix inefficiencies.”

The set with all CCR-efficient DMUs is known as “Peer set”. Other terms are used to describe these inefficiencies. “Weak efficiency” refers to condition (i) in Definition 3.2. When conditions (i) and (ii) are both satisfied, it is called “Pareto-Koopmans” or “strong” efficiency.

**Definition 3. (Pareto-Koopmans Efficiency)** A DMU is fully efficient if and only if it is not possible to improve any input or output without worsening some other input or output.

Now we set some reference set to improve the efficiency of inefficient DMU. Using the above definition (2)

**Definition 4. (Reference Set)** [8] For an inefficient DMU say DMU<sub>o</sub>, the reference set  $E_o$  is defined based on the max-slack solution obtained in phases one and two, as

$$E_o = \{j | \lambda^* > 0, j \in 1, \dots, n\}$$

An optimal solution can be expressed as the positive combination of observed input and output values. This implies that the efficiency of DMU<sub>o</sub> can be improved if the input values are reduced radially by the ratio  $\theta^*$  and input excesses recorded in  $s^-$  are eliminated. Similarly, efficiency can be attained if output values are augmented by output shortfalls in  $s^{+*}$ .

The gross input improvement  $\Delta x_o$  and output improvement  $\Delta y_o$  can be calculated using the following formulas:  $\Delta x_o = (1 - \theta^*)x_o + s^{-*}$  and  $\Delta y_o = s^{+*}$ . And we have the formula for efficiency improvement, which is also called the CCR-Projection formula defined as below

$$\widehat{\Delta x_o} = x_o - \Delta x_o = \theta^* x_o - s^{-*}$$



$$\hat{y}_o = y_o + \Delta y_o = y_o + s^{+*}$$

These relations project DMU<sub>o</sub> into the reference set  $E_o$ , and any non-negative combination of DMUs in  $E_o$  is efficient. The improved activity  $(\hat{x}_o, \hat{y}_o)$  defined by these formulas is CCR-efficient [8]. The point with coordinates  $\hat{x}_o$  and  $\hat{y}_o$ , defined by these formulas, is the point on the efficient frontier used to evaluate the performance of DMU<sub>o</sub>. [8]

In summary, the CCR projections identify the point either as a positive combination of other DMUs with  $x_o \geq \hat{x}_o$  and  $\hat{y}_o > y_o$  unless  $\theta^* = 1$  and all slacks are zero, in which case  $\hat{x}_o = x_o$  and  $\hat{y}_o = y_o$ , so the operation performed on the observation for DMU<sub>o</sub> identifies a new DMU positioned on the efficient frontier. Conversely, the point associated with the generated DMU evaluates the performance of DMU<sub>o</sub> as exhibiting input excesses  $(x_o - \hat{x}_o)$  and output shortfalls  $(\hat{y}_o - y_o)$  [8].

Till now, We have primarily discussed input-oriented models, which aim to minimize inputs while maintaining given output levels. Another model type, output-oriented models, focuses on maximizing outputs while not exceeding the observed input amounts. Output-oriented fractional program is written as follows;

$$\begin{aligned} \text{Minimize: } & \frac{\sum_{r=1}^m v_r x_{ro}}{\sum_{i=1}^s u_i y_{io}} \\ \text{Subject to: } & \frac{\sum_{r=1}^m v_r x_{rj}}{\sum_{i=1}^s u_i y_{ij}} \geq 1 (j = 1, \dots, n) \\ & \mathbf{u}, \geq 0 \quad \mathbf{v} \geq 0 \end{aligned} \tag{2.8}$$

This model is equivalent to the following LP

$$\begin{aligned} \text{Minimize: } & \sum_{r=1}^m v_r x_{ro} \\ \text{Subject to: } & \sum_{r=1}^m v_r x_{rj} - \sum_{i=1}^s u_i y_{ij} \geq 0 (j = 1, \dots, n) \\ & \sum_{i=1}^s u_i y_{io} = 1 \\ & \mathbf{u}, \geq 0 \quad \mathbf{v} \geq 0 \end{aligned} \tag{2.9}$$

And Writing it into matrix notation we get the multiplier form of Output-oriented

CCR written as follows:

$$\begin{aligned}
 & \text{[Multiplier form of Output-Oriented CCR]} \quad \text{Minimize} \quad \mathbf{v}\mathbf{x}_o \\
 & \quad \text{subject to} \quad \mathbf{u}\mathbf{y}_o = 1 \\
 & \quad \quad \quad \mathbf{u}\mathbf{Y} - \mathbf{v}\mathbf{X} \geq \mathbf{0} \\
 & \quad \quad \quad \mathbf{v} \geq 0, \mathbf{u} \geq 0
 \end{aligned} \tag{2.10}$$

Writing dual of (2.10) we get Radial/Envelopment form of the CCR output-oriented model, as follows;

$$\begin{aligned}
 & \text{[Output-Oriented CCR Radial form]} \quad \text{Max:} \quad \eta \\
 & \quad \text{Subject to:} \quad \mathbf{x}_o - \mathbf{X}\mu \geq 0 \\
 & \quad \quad \quad \eta\mathbf{y}_o - \mathbf{Y}\mu \leq 0 \\
 & \quad \quad \quad \mu \geq 0
 \end{aligned} \tag{2.11}$$

By substituting  $\eta = \frac{1}{\theta}$  and  $\mu = \eta\lambda$  in (2.11) we get the input-oriented CCR model (2.6). Hence, the optimal solution of the output-oriented model is connected to the input-oriented model's optimal solution as below:

$$\eta^* = \frac{1}{\theta^*}, \quad \mu^* = \frac{\lambda^*}{\theta^*}$$

Hence for Output oriented model efficiency will always be greater than 1. A higher value of  $\eta^*$  indicates a less efficient DMU.  $\theta^*$  represents the input reduction rate, while  $\eta^*$  denotes the output enlargement rate. It can be concluded that an input-oriented CCR model is efficient for any DMU only if it is also efficient when using the output-oriented CCR model to assess its performance. As we defined the CCR-efficiency, reference set, and CCR-Projection in the case of Input oriented model. Similarly, we can define these terms for output-oriented CCR models.

**Benifits and Shortcomings of CCR-Model:** The CCR model, enables efficiency evaluation with multiple inputs and outputs, making it a versatile and straightforward tool. However, its limitations include the assumption of constant returns to scale, sensitivity to outliers, and the inability to account for external factors. Additionally, it only measures relative efficiency and provides a snapshot of efficiency at a specific point in time, without considering dynamic changes.

## 2.1.2 BCC models

The BCC model, an extension of the CCR model, accounts for variable returns to scale, providing a more flexible analysis of efficiency. By considering production

frontiers spanned by the convex hull of existing DMUs, it enables the examination of increasing, decreasing, and constant returns to scale within the DEA framework [10].

The BCC model, introduced by Banker, Charnes, and Cooper (1984), defines the production possibility set as  $T^{(CRS)} = \{(\mathbf{x}, \mathbf{y}) | \mathbf{x} > \mathbf{X}\lambda, \mathbf{y} < \mathbf{Y}\lambda, \mathbf{e}\lambda = 1, \lambda > 0\}$ , where  $\mathbf{X}$  and  $\mathbf{Y}$  are given data sets,  $\lambda$  is a non-negative column vector, and  $\mathbf{e}$  is a row vector with all elements equal to 1. The BCC model differs from the CCR model by adding the condition  $\mathbf{e}\lambda = 1$ , which, along with the non-negativity constraint, imposes a convexity condition on how the observations for the  $n$  DMUs can be combined, enabling the analysis of variable returns to scale [8]. As we have seen different types of CCR models, we will create the following table for the different types of BCC [8] models by combining convexity constraint to do efficiency analysis for  $DMU_o$ :

	<b>Input-oriented BCC Model</b>	<b>Output-oriented BCC Model</b>
<b>Envelopment/Radial Form</b>	$\begin{aligned} &\min \theta_B \\ &\text{s.t.} \\ &\theta_B X_o - \mathbf{X}\lambda \geq \mathbf{0} \\ &\mathbf{Y}\lambda \geq Y_o \\ &\mathbf{e}\lambda = 1 \\ &\lambda \geq \mathbf{0} \end{aligned}$	$\begin{aligned} &\max \eta_B \\ &\text{s.t.} \\ &\mathbf{X}\lambda \leq x_o \\ &\eta_B y_o - \mathbf{Y}\lambda \leq \mathbf{0} \\ &\mathbf{e}\lambda \geq 0 \\ &\lambda \geq \mathbf{0} \end{aligned}$
<b>Multiplier Form</b>	$\begin{aligned} &\max z = uy_o - u_o \\ &\text{s.t.} \\ &\mathbf{v}\mathbf{x}_o = 1 \\ &-\mathbf{v}\mathbf{X} + \mathbf{u}\mathbf{Y} - u_o \leq \mathbf{0} \\ &\mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \\ &u_o, z \text{ scalars} \end{aligned}$	$\begin{aligned} &\min z = vx_o - v_o \\ &\text{s.t.} \\ &uy_o = 1 \\ &\mathbf{v}\mathbf{X} - \mathbf{u}\mathbf{Y} - v_o \mathbf{e} \geq \mathbf{0} \\ &\mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \\ &v_o, z \text{ scalars} \end{aligned}$
<b>Fractional Form</b>	$\begin{aligned} &\max \frac{uy_o - u_o}{vx_o} \\ &\text{s.t.} \\ &\frac{uy_j - u_o}{vx_j} \leq 1 (j = 1 \dots n) \\ &\mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}, u_o : \text{free} \end{aligned}$	$\begin{aligned} &\min \frac{vx_o - v_o}{uy_o} \\ &\text{s.t.} \\ &\frac{vx_j - v_o}{uy_j} \geq 1 (j = 1 \dots n) \\ &\mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}, v_o : \text{free} \end{aligned}$

Table 2.1: BCC Models [8]

Now, we define BCC-efficiency in a similar manner to how we previously defined CCR-efficiency. First, we solve the radial form of the BCC model (2.1.2) using the Two-phase method. let's assume we get the optimal solution of the Input-oriented Radial form of the BCC model (Table 2.1) as  $(\theta_B^*, \lambda^*, s^{-*}, s^{+*})$  where  $s^{-*}$  denotes

Maximal Input excesses and  $s^{+*}$  denotes maximal output shortfalls. Now we define BCC-efficiency, Reference set, and BCC-Projection similarly as of CCR models.

**Definition 5. [BCC-Efficiency]** [8] *An optimal solution  $(\theta^*, \lambda^*, s^{-*}, s^{+*})$  obtained in the two-phase process for the BCC model (BCCo) satisfies  $\theta^* = 1$  and has no slack ( $s^- = 0, s^+ = 0$ ). If these conditions are met, then the Decision-Making Unit (DMU<sub>o</sub>) is called BCC-efficient. Otherwise, it is BCC-inefficient.*

For a BCC-inefficient DMU<sub>o</sub>, we can define its reference set [8],  $E_o$ , based on an optimal solution  $\lambda^*$  by

$$E_o = \{j | \lambda_j^* > 0, \quad j \in \{1, \dots, n\}\}. \quad (2.12)$$

If multiple optimal solutions exist, we can select any of them to determine the following:

$$\theta_B^* x_o = \sum_{j \in E_o} \lambda_j^* x_j + s^{-*} \quad y_o = \sum_{j \in E_o} \lambda_j^* y_j - s^{+*} \quad (2.13)$$

Consequently, we obtain the formula for the BCC projection, as shown below. The improved activity resulting from this formula is BCC-efficient [8]:

$$\hat{x}_o \Leftarrow \theta_{\mathbf{B}}^* \mathbf{x}_o - s^{-*}, \quad \hat{y}_o \Leftarrow \mathbf{y}_o + s^{+*} \quad (2.14)$$

The BCC model offers flexibility and more accurate efficiency scores for DMUs operating under variable returns to scale, as well as better discrimination between efficient and inefficient units. However, it has shortcomings such as scale efficiency ambiguity, multiple optimal solutions, and increased complexity compared to the CCR model (Banker et al., 1984).

### 2.1.3 Additive Models

The additive program addresses the limitations of input and output orientations by simultaneously minimizing inputs and maximizing outputs, ensuring efficiency in both directions. Charnes et al. [6] introduced the Additive model by combining both input and output orientation. By considering the technology set  $T^{(VRS)}$  the additive model [6] and its dual [20] (similarly we can define additive model in  $T^{(CRS)}$ ) is given in below table:

Table 2.2: Additive model and Dual problem

Additive Model	Dual of additive Model
$\max \left( \sum_{i=1}^m s_i^- + \sum_{r=1}^s s_r^+ \right)$ <p>s.t.</p> $\sum_{j=1}^n \lambda_j x_{ij} + s_i^- = x_{i0}, \quad i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j y_{rj} - s_r^+ = y_{r0}, \quad r = 1, 2, \dots, s$ $\sum_{j=1}^n \lambda_j = 1$ $\lambda_j, s_i^-, s_r^+ \geq 0, \quad j = 1, 2, \dots, n; \quad i = 1, 2, \dots, m; \quad r = 1, 2, \dots, s$	$\max \left( \sum_{i=1}^m v_i x_{i0} - \sum_{r=1}^s u_r y_{r0} + u_0 \right)$ <p>s.t.</p> $\sum_{i=1}^m v_i x_{ij} - \sum_{r=1}^s u_r y_{rj} + u_0 \geq 0, \quad j = 1, 2, \dots, n$ $v_i \geq 1, \quad i = 1, \dots, m$ $u_r \geq 1, \quad r = 1, 2, \dots, s$ $u_0 \text{ free}$

This Additive model is always feasible ( $\lambda_o = 1$  and  $\lambda_j = 0$  for all  $j \neq o, s^- = s^+ = 0$ ) and bounded. This model uses input excesses and output shortfalls simultaneously to arrive at efficiency frontier. We have the following definition for efficiency in the case of the Additive model;

**Definition 6 (ADD-efficient DMU).** *Let the optimal solution of  $DMU_o$  is  $(\lambda^*, \mathbf{s}^{-*}, \mathbf{s}^{+*})$ , then  $DMU_o$  is efficient  $\iff$  slacks in optimal solution is equal to zero.*

All the constraints of the Additive model (2.2) are constraints of the BCC model in table (2.1), from here we can observe that efficient DMUs in both models (BCC and Additive) are exactly same. And for inefficient DMUs, we improve them to efficient DMUs by projection formula ( $\widehat{x}_{io} \leftarrow x_{io} - s_i^{-*}$   $\widehat{y}_{ro} \leftarrow y_{ro} + s_r^{-*}$ ), and this improved activity is ADD-efficient (5). One of the main advantages of this model is that it remains invariant (7) under the translation of inputs and outputs data. One major shortcoming of this model is that it is not invariant under units of inputs and outputs because of the summation of slacks in the objective function. To tackle this problem Tone (18) have introduced a measure called ‘‘Slack-Based Measure (SBM)’’, which I will discuss in the next part.

## 2.1.4 SBM Models

The Slacks-Based Measure (SBM) is a dimension-free, units-invariant efficiency evaluation method for additive models. Introduced by Tone, it provides a single scalar

that remains consistent across varying units of measurement for inputs and outputs. This section introduces a single scalar called "SBM" (Slacks-Based Measure) for additive models, which was introduced by Tone [18] and possessed important properties like unit invariance and monotonicity. SBM is unit-invariant and ensures a monotone decrease in each input and output slack. To measure the efficiency we have the following fractional program which can be converted into LP [18] using the methods which we used in CCR models;

$$\min \rho = \frac{1 - \frac{\sum_{i=1}^m \frac{s_i^-}{x_{i0}}}{m}}{1 + \frac{\sum_{j=1}^s \frac{s_r^+}{y_{r0}}}{s}}$$

subject to:

$$\sum_{j=1}^n \lambda_j x_{ij} = x_{i0} - s_i^-, \quad i = 1, 2, \dots, m \quad (2.15)$$

$$\sum_{j=1}^n \lambda_j y_{rj} = y_{r0} + s_r^+, \quad r = 1, 2, \dots, s$$

$$\lambda_j \geq 0, \quad j = 1, 2, \dots, n$$

$$s_i^- \geq 0, i = 1, 2, \dots, m, s_r^+ \geq 0, \quad r = 1, 2, \dots, s$$

After converting it into an LP program, we get the following model with different transformed decision variables;

$$\min \tau = t - \frac{\sum_{i=1}^m \frac{S_i^-}{x_{i0}}}{m}$$

subject to:

$$t + \frac{\sum_{j=1}^s \frac{S_r^+}{y_{r0}}}{s} = 1$$

$$\sum_{j=1}^n \Lambda_j x_{ij} = t x_{i0} - S_i^-, \quad i = 1, 2, \dots, m \quad (2.16)$$

$$\sum_{j=1}^n \Lambda_j y_{rj} = t y_{r0} + S_r^+, \quad r = 1, 2, \dots, s$$

$$\Lambda_j \geq 0, \quad j = 1, 2, \dots, n; t \geq 0$$

$$S_i^- \geq 0, i = 1, 2, \dots, m \quad S_r^+ \geq 0, r = 1, 2, \dots, s$$

Let the optimal solution of model (2.16) is  $(\tau^*, \Lambda^*, \mathbf{S}^-, \mathbf{S}^+, t^*)$  and optimal solution of model (2.15) is  $(\rho^*, \lambda^*, \mathbf{s}^-, \mathbf{s}^+)$ , Then these solution is related by following equations [18];

$$\rho^* = \tau^*; \quad \lambda^* = \frac{\Lambda^*}{t^*}; \quad s^{-*} = \frac{S^{-*}}{t^*}; \quad s^{+*} = \frac{S^{+*}}{t^*}$$

In the case of Slack-based measures, we say  $DMU_o$  is SBM-efficient if  $\rho^* = 1$  (equivalent to saying all input and output slacks is zero). And for inefficient  $DMU_o$  we make it efficient by below equation;

$$x_{io} \leftarrow x_{io} - s_i^{-*} \quad y_{ro} \leftarrow y_{ro} + sr^{+*}$$

$$i = 1, 2, \dots, m; \quad r = 1, 2, \dots, s$$

, which is known as SBM-Projection [5]. And we define the Reference set [18] of  $DMU_o$  by

$$sbmR_o = \{j; \lambda_j^* > 0, j = 1, 2, \dots, n\}$$

The Slacks-Based Measure (SBM)  $\rho^*$  focuses on reference-set dependent values, making it distinct from other efficiency measures that use the entire data set. This targeted evaluation is not influenced by the performance of unrelated Decision-Making Units (DMUs). We have the following results which is proved in the paper [18] which relate SBM and CCR models ;

- (1)  $0 \geq \rho^* \leq \theta_{CCR}^*$ , and (2)  $DMU_o$  is SBM-efficient  $\Leftrightarrow$  it is CCR-efficient.

**Benefits and shortcomings;** The Slacks-Based Measure (SBM) presents advantages like unit invariance, monotonicity concerning slacks, and dependency on reference sets, distinguishing it from other efficiency measures such as CCR and BCC. Nonetheless, the additive model can differentiate between efficient and inefficient DMUs, but it falls short in assessing the extent of inefficiency itself, which can be viewed as a limitation.

## 2.2 Conclusions

This chapter has highlighted the importance and relevance of DEA models in evaluating DMUs' efficiency. We have discussed the motivation behind developing DEA models and their fundamental concepts, such as input/output vectors and production

possibility sets. We also introduced the most widely used DEA models, including CCR, BCC, additive models, and SBM, and explained their benefits and limitations.

This chapter provides a solid foundation for further exploring DEA models' applications and their use in various industries. By understanding the different models and their assumptions, we can select the most appropriate model for their research and gain insights into improving DMU's efficiency.

Furthermore, this chapter has highlighted the significance of DEA models as a non-parametric approach to evaluating DMUs' efficiency, providing insights that could aid in improving their performance. Exploring the applications of DEA models can be valuable for decision-makers seeking to make informed decisions and improve performance.



# Chapter 3

## Inverse DEA

### Introduction

The effective allocation of resources is a crucial aspect of any decision-making process. In the context of organizations, this entails the efficient distribution of inputs and outputs to achieve maximum performance. Data Envelopment Analysis (DEA) has emerged as a widely used method for evaluating the relative efficiency of decision-making units (DMUs) by comparing their inputs and outputs. While classical DEA provides valuable insights, it does not directly address the question of how to adjust inputs and outputs to attain a desired efficiency level. This leads to the development of the inverse DEA approach.

Inverse DEA is an extension of classical DEA that focuses on determining the necessary changes in input and output levels to reach a target efficiency score. This approach provides a valuable tool for managers and policymakers, helping them understand the magnitude of adjustments required to attain their desired efficiency. Inverse DEA models are particularly useful in budgeting and planning scenarios, where managers need to make decisions about resource allocation to achieve specific efficiency goals.

This chapter aims to introduce the concept of inverse DEA, drawing inspiration from the paper "New Inverse DEA Models for Budgeting and Planning" by Sayar et al. ([16]). We will start with the classical inverse DEA model, then explore income-based and budget-based inverse DEA models, highlighting their applications in various settings. The objective is to provide a comprehensive understanding of the inverse DEA approach and its practical implications for organizational decision-making processes.

Following this introduction, we will delve into the classical inverse DEA model, as

presented in Sayar et al.'s paper. We will then explore income-based and budget-based inverse DEA models (16), illustrating their use in real-world scenarios. The python code for solving these models will be provided in the appendix section of this thesis. By the end of this chapter, we will have a solid understanding of inverse DEA's relevance and potential applications in budgeting and planning contexts, as well as a deeper appreciation for the contributions made by Sayar et al. in this field.

### 3.1 Classical Inverse DEA Models

Following the notations as on the Notation page (0.1), we have two types of models for efficiency calculations one is input-oriented and the other is output-oriented with the different return to scale denoted by  $\mathbf{S}(p) : p \in \{CRS, VRS, NIRS, NDRS\}$  (2.2). Both Models are given below in the table for calculating the efficiency of the target  $DMU_o$

Table 3.1: Generalized Input-oriented and output-oriented DEA models

Model	Input-oriented	Output-oriented
Objective	$\theta_o = \min \theta$	$\phi_o = \max \phi$
Subject to	$\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io} \quad \forall i = 1, \dots, m$	$\sum_{j=1}^n \lambda_j x_{ij} \leq x_{io} \quad \forall i = 1, \dots, m$
	$\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro}; \forall r = 1, \dots, m$	$\sum_{j=1}^n \lambda_j y_{rj} \geq \phi y_{ro}; \forall r = 1, \dots, m$
	$\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbf{S}(p)$	$\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbf{S}(p)$

As inverse DEA aims to answer questions such as: If, among a group of DMUs, we increase certain inputs to a particular unit and assume that the DMU maintains its current efficiency level with respect to other units, how much more output could the unit produce? Alternatively, if the outputs need to be increased to a certain level and the unit's efficiency remains unchanged, how much more input should be provided to the unit? These types of questions are generally addressed using Multiple Objective Linear Programming (MOLP) in the inverse DEA literature. Let's assume we have two scenarios in which we perturb the output and input levels of  $DMU_o$ .

1. In the first scenario, with an input-oriented perspective, we perturb the output level of  $DMU_o$  from  $y_o$  to  $(y_o + \Delta y_o)$ . The classical inverse DEA models aim to

find the level of the required inputs needed for producing the new output level  $(y_o + \Delta y_o)$  while maintaining the current efficiency level. The below table for this scenario illustrates the impact of perturbing output levels on the required inputs. This formulation is a resource allocation problem.

2. In the second scenario, with an output-oriented perspective, assume we perturb the input levels of DMU<sub>o</sub> from  $x_o$  to  $(x_o + \Delta x_o)$ . The classical inverse DEA model seeks to determine the producible output level (while maintaining the current efficiency measure) when the input level is perturbed to  $(x_o + \Delta x_o)$ . The below table for this scenario shows the impact of perturbing input levels on the producible outputs. This formulation is regarded as a forecasting problem.

By examining these two scenarios we can analyze how the efficiency of DMUs is affected by perturbations in input and output levels in both input-oriented and output-oriented contexts. This provides valuable insights into the performance of each DMU under these new conditions and helps in better decision-making for resource allocation and performance improvement.

Let analyse classical Inverse DEA in case of output-oriented model, first we will introduce a general model for Output-oriented model for different return-to-scale as follows:

$$\begin{aligned}
(E_o)\phi_o &= \max \phi \\
\text{s.t.} & \\
\sum_{j=1}^n \lambda_j x_{ij} &\leq x_{io}, \forall i = \{1, \dots, m\} \\
\sum_{j=1}^n \lambda_j y_{rj} &\geq \phi y_{r0}, r = \{1, \dots, s\} \\
\delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) &= \delta_1, \\
\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i &\in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.1}$$

The dual problem for above model (3.1) given as ;

$$\begin{aligned}
(D_o) \max z &= \sum_{i=1}^m w_i x_{io} + \tau \delta_1 \\
\text{Subject to} \\
\sum_{i=1}^m w_i x_{ij} - \sum_{r=1}^s u_r y_{rj} + \delta_1 \tau &\geq, \forall j = \{1, \dots, n\} \\
\sum_{r=1}^s u_r y_{ro} &= 1 \\
w_i \geq 0, \forall i = \{1, \dots, m\} \quad u_r &\geq 0, \forall r = \{1, \dots, s\} \\
\delta_1 \delta_2 (-1)^{\delta_3} \tau &\geq 0, \tau \text{ is free}
\end{aligned} \tag{3.2}$$

As depending on different value of set  $\{\delta_1, \delta_2, \delta_3\}$  we have different type of model, let  $\{\delta_1 = 1, \delta_2 = 0, \delta_3 = 0\}$  then we get VRS(BCC model) model. and similiary we can get CRS, NIRS, NDRS model for different set of  $\{\delta_1, \delta_2, \delta_3\}$ .

**Definition 7.** [19] The optimal value  $\phi_0$  of problem (3.1), under a particular set of values of  $\delta_1, \delta_2$  and  $\delta_3$ , is called the efficiency index of  $DMU_0$  under the corresponding DEA model. If the efficiency index  $\phi_0 = 1$ , then we say the  $DMU_0$  is (at least) weakly DEA efficient under the particular model.

Now considering the second scenario, purturbing the input level of  $DMU_o$  from  $x_o = (x_{1o}, \dots, x_{mo})$  to  $x_o + \Delta x_o = (x_{1o} + \Delta x_{1o}, \dots, x_{mo} + \Delta x_{mo})$  where  $\Delta x_o \geq$  and  $\Delta x_o \neq 0$ . Then we need to find how much output vector purtubed by keeping the efficiency index same(i.e.  $\phi_o$ ). let  $y_o = (y_{1o}, \dots, y_{so})$  increased to  $y_o + \Delta y_o = (y_{1o} + \Delta y_{1o}, \dots, y_{so} + \Delta y_{so})$  then need question is to evaluate  $\Delta y_o$ . To do this let  $DMU_{n+1}$  represent  $DMU_o$  after perturbation in inputs. Then corresponding output DEA model for measuring the efficiency of  $DMU_{n+1}$  can be represented using the following model;

$$\begin{aligned}
(E_o^+) \phi_{n+1} &= \max \phi \\
\text{s.t.} & \\
\sum_{j=1}^n \lambda_j x_{ij} + \lambda_{n+1}(x_{io} + \Delta x_{io}) &\leq (x_{io} + \Delta x_{io}), \forall i = \{1, \dots, m\} \\
\sum_{j=1}^n \lambda_j y_{rj} + \lambda_{n+1}(y_{ro} + \Delta y_{ro}) &\geq \phi(y_{ro} + \Delta y_{ro}), r = \{1, \dots, s\} \\
\delta_1 \left( \sum_{j=1}^n \lambda_j + \lambda_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) &= \delta_1, \\
\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.3}$$

Now by considering following MOLP(multi Objective LP) with optimal value  $\phi_o$  of [\(3.1\)](#) ;

$$\begin{aligned}
(MOLP - O) \phi_{n+1} &= \max(\Delta y_{1o}, \dots, \Delta y_{so}) \\
\text{s.t.} & \\
\sum_{j=1}^n \lambda_j x_{ij} + \lambda_{n+1}(x_{io} + \Delta x_{io}) &\leq (x_{io} + \Delta x_{io}), \forall i = \{1, \dots, m\} \\
\sum_{j=1}^n \lambda_j y_{rj} + \lambda_{n+1}(y_{ro} + \Delta y_{ro}) &\geq \phi_o(y_{ro} + \Delta y_{ro}), r = \{1, \dots, s\} \\
\delta_1 \left( \sum_{j=1}^n \lambda_j + \lambda_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) &= \delta_1, \\
\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \Delta y_{ro} \geq 0; r = 1, \dots, s \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.4}$$

Then for above MOLP model we define a Weak-Pareto solution as follows;

**Definition 8** (Weak-Pareto solution). [\[19\]](#) Let  $(\Delta \bar{y}_o, \bar{\lambda}, \bar{\nu}) \in \mathbb{R}^s \times \mathbb{R}^t \times \mathbb{R}$  be a feasible solution of the MOLP [\(3.4\)](#), we call this feasible solution Weak-Pareto solution if there is no other feasible solution that performs better in all objectives simultaneously, i.e. there is no other feasible solution  $(\Delta y_o, \lambda, \nu)$  of the MOLP [\(3.4\)](#) such that  $\Delta y_o > \Delta \bar{y}_o$ .

Now we have the following lemma's and Theorems established by Wei et al. [\[19\]](#):

**lemma 1.** Suppose  $(\bar{\Delta y}_o, \bar{\lambda}, \bar{\nu})$  is a Weak-Pareto solution of MOLP(3.4) and  $\phi_o$  be optimal solution of (3.1) then optimal value of following model is  $\phi_o$ .

$$\begin{aligned}
& (\bar{E}_o) \max \phi \\
& \text{s.t.} \\
& \sum_{j=1}^n \lambda_j x_{ij} \leq x_{io} + \Delta x_{io}, \forall i = \{1, \dots, m\} \\
& \sum_{j=1}^n \lambda_j y_{rj} \geq \phi(y_{r0} + \Delta \bar{y}_{ro}), r = \{1, \dots, s\} \\
& \delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& \nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.5}$$

**lemma 2.** Given a feasible solution  $(\bar{\Delta y}_o, \bar{\lambda}, \bar{\nu})$  of MOLP(3.4), If optimal value of problem(3.5) is  $\phi_o$ . then  $(\bar{\Delta y}_o, \bar{\lambda}, \bar{\nu})$  is a Weak-Pareto solution of MOLP(3.4).

Define an another model as below by introducing the term  $\lambda_{n+1}$  in(3.5);

$$\begin{aligned}
& (\bar{E}_o+) \max \phi \\
& \text{s.t.} \\
& \sum_{j=1}^n \lambda_j x_{ij} + \lambda_{n+1}(x_{io} + \Delta x_{io}) \leq x_{io} + \Delta x_{io}, \forall i = \{1, \dots, m\} \\
& \sum_{j=1}^n \lambda_j y_{rj} + \lambda_{n+1}(y_{io} + \Delta \bar{y}_{io}) \geq \phi(y_{r0} + \Delta \bar{y}_{ro}), r = \{1, \dots, s\} \\
& \delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& \nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.6}$$

**lemma 3.** Let we have  $\Delta y_o \geq 0$ . Then If optimal value for (3.6) is  $\phi_o$  then optimal value of(3.5) is also  $\phi_o$ . converse is also true.

**Theorem 2.** Assume DMU<sub>o</sub> have efficiency score  $\phi_o$  for(3.1) and its input will increase from  $x_o$  to  $x_o + \Delta x_o$  with  $\Delta x_o \geq 0, \Delta x_o \neq 0$  then; If  $(\bar{\Delta y}_o, \bar{\lambda}, \bar{\nu})$  is a Weak-Pareto solution of MOLP(3.4) then by increasing the outputs by  $\Delta \bar{y}_o$ , the new DEA

problem (3.6) have same efficiency score  $\phi_o$ . Conversely if  $(\bar{\Delta}y_o, \bar{\lambda}, \bar{\nu})$  is a feasible solution of MOLP (3.4) and if optimal value of (3.6) is  $\phi_o$  the  $(\bar{\Delta}y_o, \bar{\lambda}, \bar{\nu})$  is weak pareto solution of MOLP (3.4).

Now we know that Inverse DEA problem can be converted into a MOLP. Now we show that in certain cases we need to solve only a single-objective LP problem to get more definite answer to Inverse DEA problem. Consider the scenario where each output has a known per-unit weight such as income, profit or a specific weight assigned to each output. Let  $a_r$  denotes the value of weights for each output unit  $r$  with  $a_r > 0$  then by considering the following LP problem,

$$\begin{aligned}
& \text{maximize} && \sum_{r=1}^s a_r \Delta y_{ro} \\
& \text{subject to:} && \\
& \sum_{j=1}^n \lambda_j x_{ij} \leq (x_{io} + \Delta x_{io}), && i = 1, \dots, m \\
& \sum_{j=1}^n \lambda_j y_{rj} \geq \phi_o (y_{ro} + \Delta y_{ro}), && r = 1, \dots, s \\
& \delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& \nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{3.7}$$

Any optimal solutions of above LP problem must be a weak-pareto solution of the MLOP (3.4) [19].

**Corollary:** Suppose  $DMU_o$  has an efficiency index  $\phi_o$  under the model (3.1) and input is increased by  $\delta x_o$ . If  $(\bar{\Delta}y_o, \bar{\lambda}, \bar{\nu})$  is an optimal solution of (3.7) then by increasing the output by  $\bar{\Delta}y_o$ , efficiency index of  $DMU_o$  remains same under new inputs and outputs. [19].

Similar analysis will work for first scenario, in case of input-oriented model table (C.1) shows MOLP form of inverse DEA in case of Input and output orientation.

Now Based on both scenario, I will explain Income and Budget based inverse DEA with applications which is work of SAYAR [16].

## 3.2 Inverse DEA for Budgeting and Planning

In this section, we will explore two types of inverse DEA models that build upon the concept of inverse DEA. The first model is the income-based model with an input orientation, and the second model is the budget-based model with an output orientation. These models are designed to determine how changes in income and budget affect the efficiency of decision-making units.

### 3.2.1 Income-based Inverse DEA Model

In the income-based inverse DEA approach, the decision-maker (DM) aims to increase the total income of a specific DMU<sub>o</sub> in order to reach a minimum income level denoted by  $A$ . The primary objective is to determine the necessary input adjustments required to achieve this target while preserving the current efficiency level. The following model is formulated to address this objective [16]:

$$\begin{aligned}
 & \text{minimize } \Delta x_o = (\Delta x_{1o}, \Delta x_{2o}, \dots, \Delta x_{mo}) \\
 & \text{subject to:} \\
 & \sum_{r=1}^s a_r * \Delta y_{ro} \geq A \\
 & \sum_{j=1}^n \lambda_j * x_{ij} \leq \theta_o * (x_{io} + \Delta x_{io}), \quad i = 1, \dots, m \\
 & \sum_{j=1}^n \lambda_j * y_{rj} \geq (y_{ro} + \Delta y_{ro}), \quad r = 1, \dots, s \\
 & \lambda = (\lambda_1, \dots, \lambda_n) \in S(p)
 \end{aligned} \tag{3.8}$$

- Above Model (3.8) minimizes the necessary input adjustments  $(\Delta x_{1o}, \Delta x_{2o}, \dots, \Delta x_{mo})$  to reach the desired income level  $A$ .
- The model considers the price of the  $r$ th output as  $a_r$  ( $1 \leq r \leq s$ ).



- The model finds the minimum input level required to produce the desired output level and achieve the minimum total income of DMU<sub>o</sub> (A) while maintaining the current efficiency level.
- $\Delta x_0 \in \mathbb{R}_+^m$ ,  $\Delta y_0 \in \mathbb{R}_+^s$ , and  $\lambda \in \mathbb{R}_+^n$  are decision variables in the model.
- The associated classical inverse DEA model (C.1) can be considered a special case of the income-based inverse DEA model (3.8).

Key differences between the income-based inverse DEA model (3.8) and the classical inverse DEA model are as follows:

- The classical inverse DEA model focuses on achieving individual output levels, while the income-based inverse DEA model (3.8) is concerned with reaching a total income level with the minimum level of inputs.
- $\Delta y_{r_0}$  represents the expected value of the output and serves as a vector parameter in the classical model.
- Both input and output levels are variable in the income-based inverse DEA model (3.8), while the output level is given and acts as a parameter in the associated inverse model.

The classical inverse DEA model (Input-Oriented) (C.1) is a special case of the income-based model (3.8). The income-based model is more generalized and flexible than its classical counterpart. By considering  $\Delta y_{r_0} = \Delta \bar{y}_{r_0}$  and ignoring the income constraint in the income-based model, the classical inverse DEA model can be derived.

As we discussed in the section on classical inverse DEA, the same pattern will work here, we also have definition of the Weak-Pareto solution in the case of the Income-Based model (3.8) as follows.

**Definition (Weak-pareto Solution):** Suppose  $(\lambda, x_0 + \Delta x_0, y_0 + \Delta y_0)$  is a feasible solution for model (3.8). If there is no feasible solution  $(\lambda, x_0 + \Delta x'_0, y_0 + \Delta y'_0)$  of this model such that  $x_0 + \Delta x'_0, y_0 < x_0 + \Delta x_0, y_0 + \Delta y_0$ , then we say the  $(\lambda, x_0 + \Delta x_0, y_0 + \Delta y_0)$  is a Weak-Pareto solution of model (3.8). [16]

**Theorem (Efficiency Preservation):** If  $(\lambda, x_0 + \Delta x_0, y_0 + \Delta y_0)$  is a weak efficient solution of income-based model (3.8), then the efficiency of  $DMU_o = (x_0, y_0)$  and  $DMU_{o+} = (x_0 + \Delta x_0, y_0 + \Delta y_0)$  are the same. [16]

As the model (3.8) is a multiple objective linear programming (MOLP) problem and does not have an optimal solution; instead, it has an efficient solution. As we have discussed in the previous section, in similar fashion by Considering the input weights, we can use the weighted sum approach to solve the MOLP problem (3.8) and subsequently obtain the following linear programming problem:

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^m c_i \Delta x_{i0} \\
& \text{subject to} && \sum_{r=1}^s a_r \Delta y_{r0} \geq A \\
& && \sum_{j=1}^n \lambda_j x_{ij} \leq \rho_0 (x_{i0} + \Delta x_{i0}) \quad i = 1, \dots, m \\
& && \sum_{j=1}^n \lambda_j y_{rj} \geq (y_{r0} + \Delta y_{r0}) \quad r = 1, \dots, s \\
& && \sum_{j=1}^n \lambda_j = 1 \\
& && \lambda \geq 0
\end{aligned} \tag{3.9}$$

where  $c_i \geq 0$ ,  $1 \leq i \leq m$ , is the price of the  $i$ -th input. For this LP formulation of MLOP (3.8) we have the following theorem in a similar manner of the previous section.

**Theorem 3.** *When a solution  $(\bar{\lambda}, \bar{\Delta}x, \bar{\Delta}y)$  is optimal for the linear programming model (3.9), it also qualifies as a weak-Pareto solution for the MOLP model (3.8).*

**Remark:** The optimal solution of (3.9) preserves the efficiency score of the unit under evaluation. If  $(\bar{\lambda}, \bar{\Delta}x, \bar{\Delta}y)$  is an optimal solution of the LP model (3.9), then the efficiency of  $DMU_o$  remains the same before and after adjusting its input and output levels, i.e.,  $DMU_o = (x_0, y_0)$  and  $DMU_{o+} = (x_0 + \bar{\Delta}x, y_0 + \bar{\Delta}y)$ .

### 3.2.2 Budget-based inverse DEA model

In the budget-based inverse DEA approach, the decision-maker (DM) aims to maximize the total output increment of a specific DMU<sub>o</sub> while having a given budget constraint, denoted by B. The primary objective is to determine the necessary input and output adjustments required to achieve this target while preserving the current efficiency level. The following model is formulated to address this objective:

$$\begin{aligned}
 & \text{Maximize } \Delta y_0 = (\Delta y_{1o}, \Delta y_{2o}, \dots, \Delta y_{so}) \\
 & \text{subject to:} \\
 & \sum_{i=1}^m b_i \Delta x_{io} \leq B \\
 & \sum_{j=1}^n \lambda_j x_{ij} \leq (x_{io} + \Delta x_{io}), \quad i = 1, \dots, m \\
 & \sum_{j=1}^n \lambda_j y_{rj} \geq \phi_o(y_{ro} + \Delta y_{ro}), \quad r = 1, \dots, s \\
 & \lambda = (\lambda_1, \dots, \lambda_n) \in S(p)
 \end{aligned} \tag{3.10}$$

- Above Model (3.10) maximizes the total output increment  $(\Delta y_{1o}, \Delta y_{2o}, \dots, \Delta y_{so})$  under the given budget constraint B.
- The model considers the price of the i-th input as  $b_i (1 \leq i \leq m)$ .
- The model finds the necessary input and output adjustments required to achieve the maximum output increment while maintaining the current efficiency level.
- $\Delta x_0 \in R_+^m$ ,  $\Delta y_0 \in R_+^s$ , and  $\lambda \in R_+^n$  are decision variables in the model.

Key differences between the budget-based inverse DEA model (3.10) and the income-based inverse DEA model (C.1) are as follows:

- The budget-based inverse DEA model focuses on maximizing the total output increment, while the classical inverse DEA model is concerned with achieving individual output levels with a given set of inputs.

- $\Delta x_{i0}$  represents the expected value of the input and serves as a vector parameter in the classical model.
- Both input and output levels are variable in the budget-based inverse DEA model, whereas in the classical inverse DEA model, the output level is given and acts as a parameter.

Theorem (Efficiency Preservation): If  $(\lambda, x_0 + \Delta x_0, y_0 + \Delta y_0)$  is a weak-Pareto solution of the budget-based model (3.10), then the efficiency of  $DMU_o = (x_0, y_0)$  and  $DMU_{o+} = (x_0 + \Delta x_0, y_0 + \Delta y_0)$  are the same.

The model (3.10) is a multiple objective linear programming (MOLP) problem and does not have an optimal solution; instead, it has an efficient solution. By considering the output weights, we can use the weighted sum approach to solve the MOLP problem (3.10) and subsequently obtain the following linear programming problem as we have in Classical Inverse DEA section:

$$\begin{aligned}
& \text{maximize} && \sum_{r=1}^s a_r \Delta y_{ro} \\
& \text{subject to:} && \\
& && \sum_{i=1}^m b_i \Delta x_{io} \leq B \\
& && \sum_{j=1}^n \lambda_j x_{ij} \leq \rho_0 (x_{io} + \Delta x_{io}), \quad i = 1, \dots, m \\
& && \sum_{j=1}^n \lambda_j y_{rj} \geq (y_{ro} + \Delta y_{ro}), \quad r = 1, \dots, s \\
& && \lambda \geq 0
\end{aligned} \tag{3.11}$$

where  $a_r \geq 0$ ,  $1 \leq r \leq s$ , is the weight of the r-th output. For this LP formulation of MLOP (3.10), we have the following theorem in a similar manner to the previous section.

**Theorem 4.** *When a solution  $(\bar{\lambda}, \bar{\Delta}x, \bar{\Delta}y)$  is optimal for the linear programming model (3.11), it also qualifies as a weak-Pareto solution for the MOLP model (3.10).*

Remark: The optimal solution of (??) preserves the efficiency score of the unit under evaluation. If  $(\bar{\lambda}, \bar{\Delta}x, \bar{\Delta}y)$  is an optimal solution of the LP model (??), then the efficiency of  $DMU_o$  remains the same before and after adjusting its input and output levels, i.e.,  $DMU_o = (x_0, y_0)$  and  $DMU_{o+} = (x_0 + \bar{\Delta}x, y_0 + \bar{\Delta}y)$ .

### 3.3 Applications of Income-based and Budget-based Models

In this section, we apply the income-based and budget-based inverse DEA models to a hypothetical dataset representing a group of retail stores. By examining the efficiency of these stores in terms of their resource utilization and output generation, we aim to demonstrate the practical application of these models for decision-making and strategic planning. The insights derived from this analysis can help store managers and stakeholders identify areas for improvement and optimize resource allocation, ultimately leading to enhanced performance and customer satisfaction.

The below sections explain the hypothetical data that I have assumed;

#### Data Description

The hypothetical data represents 15 stores and their input and output values.

#### Input Data

For the input data:

1. Store Area (in square meters) - This represents the size of the store. A larger store may require more resources, such as more employees or higher utility costs, to maintain and operate.
2. Man-hours - This represents the total number of hours worked by all employees in the store. It indicates the labor input necessary to run the store and provide customer services.

## Output Data

For the output data:

1. Revenue (in Indian Rupees) - This represents the total income generated by the store. Higher revenue indicates better financial performance and a more successful store.
2. Customer Loyalty (a score between 0 and 1) - This represents loyalty and satisfaction among the store's customers. A higher score means customers are more likely to return to the store and recommend it to others. This is an important non-financial indicator of store performance.

The table below shows data of all stores with their input and output efficiencies;

Table 3.2: Store Data with Input and Output Efficiencies

Store	Area (sq.m)	Man-hours	Revenue (INR)	Customer Loyalty	Input Efficiency	Output Efficiency
1	500	1200	500000	0.80	1.000	1.000
2	600	1500	600000	0.82	0.951	1.012
3	450	1000	400000	0.75	1.000	1.000
4	800	2000	900000	0.85	0.958	1.036
5	900	2200	1000000	0.90	0.947	1.022
6	400	1100	350000	0.70	0.882	1.067
7	550	1300	450000	0.78	0.865	1.042
8	650	1700	700000	0.84	0.986	1.006
9	300	900	300000	0.68	1.000	1.000
10	750	1800	800000	0.83	0.950	1.054
11	700	1600	750000	0.81	1.000	1.000
12	850	2100	950000	0.88	0.952	1.028
13	350	950	320000	0.72	1.000	1.000
14	950	2300	1100000	0.92	0.971	1.016
15	1000	2400	1200000	0.95	1.000	1.000

To assign reasonable input and output prices, we consider the approximate costs and values associated with each input and output in the Indian Currency.

For the input prices:

1. Store Area (in square meters): The cost of maintaining and operating a store per square meter can vary depending on factors such as location, utility costs, and rent. We assume an average cost of INR 2,000 per square meter per month.

2. Man-hours: The cost of labor per hour can also vary depending on the skill and experience of the employees. We assume an average cost of INR 200 per man-hour.

For the output prices:

1. Revenue (in Indian Rupees): Since revenue is already in monetary terms (Indian Rupees), we assign a value of 1 to convert revenue to the same unit.
2. Customer Loyalty (a score between 0 and 1): Assigning a monetary value to customer loyalty can be challenging, as its impact on revenue and long-term success may not be directly proportional. However, we assume that an increase of 0.1 in customer loyalty score translates to an additional INR 10,000 in revenue per month.

### Analysis of Income based inverse DEA:

In the income-based inverse DEA model, we aimed to increase the income by INR 80,000 for each store while analyzing the required changes in inputs and outputs. The results from the model are shown in the following table and their python code listed in Appendix section;

Table 3.3: Results after applying income-based model with A=80,000

Stores	$\Delta x_1$ (Change in Store Area)	$\Delta x_2$ (Change in Man-hours)	$\Delta y_1$ (Change in Revenue (INR))	$\Delta y_2$ (Change in Customer Loyalty (0-1))
1	44.444444	146.666667	80000.000000	0.000000
2	46.753247	97.792208	80000.000000	0.000000
3	1.851852	191.111111	80000.000000	0.000000
4	64.652420	121.149595	79661.016949	0.033898
5	57.798344	124.102042	79860.418744	0.013958
6	54.613858	165.721849	79810.568295	0.018943
7	30.266204	163.940972	80000.000000	0.000000
8	45.086705	15.549133	80000.000000	0.000000
9	62.036114	132.934530	79760.717846	0.023928
10	40.447371	164.492487	79760.717846	0.023928
11	11.997341	182.851446	79710.867398	0.028913
12	61.309778	122.894931	79810.568295	0.018943
13	35.185185	111.111111	80000.000000	0.000000
14	63.716124	137.087702	79760.717846	0.023928
15	NaN	NaN	NaN	NaN

The results in the above table show how each store should adjust its inputs and outputs to achieve the targeted increase in revenue. Note that some stores require no change in customer loyalty (change in customer loyalty is 0) to achieve the desired

increase in income, while others need a slight improvement in customer loyalty along with changes in store area and man-hours.

For instance, Store 1 needs to increase its store area by 44.444 sq.m and man-hours by 146.667 to achieve an income increase of INR 80,000 without any change in customer loyalty. On the other hand, Store 4 needs to increase its store area by 64.652 sq.m and man-hours by 121.150 to achieve a revenue increase of INR 79,661.017 while also improving its customer loyalty by 0.033898. Now for store 15 analysis; It is possible that Store 15 is already operating efficiently, with no suggested improvements from the income-based inverse DEA model. To better evaluate Store 15's performance, we can consider exploring alternative evaluation methods, such as incorporating additional variables or comparing it to similar stores outside the dataset to reveal potential areas for improvement.

The below table shows the behavior of store 15 with different income levels;

Table 3.4: Analysis of Store 13 for Different Income Levels

Income Level (A)	$\Delta x_1$ (Change in Store Area)	$\Delta x_2$ (Change in Man-hours)	$\Delta y_1$ (Change in Revenue (INR))	$\Delta y_2$ (Change in Customer Loyalty (0-1))
100,000	46.296296	147.777778	100,000.000000	0.000000
300,000	198.454636	482.402792	299,441.674975	0.055833
500,000	353.544921	814.739116	498,843.469591	0.115653
700,000	508.635205	1,147.075440	698,245.264207	0.175474
800,000	586.180348	1,313.243603	797,946.161515	0.205384
900,000	NaN	NaN	NaN	NaN

From the table, we can observe that as the income level (A) increases, both the required store area  $\Delta x_1$  and man-hours ( $\Delta x_2$ ) increase. This indicates that to achieve higher revenue levels, the store needs to allocate more resources in terms of area and labor. The customer loyalty ( $\Delta y_2$ ) also improves as the income level increases. It is clear from the above table is that, at an income level of 900,000, the model could not find any feasible solution, suggesting that the store may have reached its maximum potential with the given resources, and further improvement may not be possible.

#### **Analysis of Budget-based inverse DEA:**

In the budget-based inverse DEA model, we allocated a budget of INR 80,000 to each store to improve their performance and analyzed the required changes in inputs and outputs. The results obtained from the model are presented in the table below, and the corresponding Python code can be found in the Appendix section:

In the results of the budget-based model with a budget (B) of 80,000, we can observe a variety of patterns and differences across the 15 stores.



Table 3.5: Results of the budget-based model with B=80,000

Stores	$\Delta x_1$ (Change in Store Area)	$\Delta x_2$ (Change in Man-hours)	$\Delta y_1$ (Change in Revenue (INR))	$\Delta y_2$ (Change in Customer Loyalty (0-1))
1	30.075188	99.248120	54,135.338346	0.000000
2	34.586466	54.135338	93,794.727783	0.000000
3	16.165414	238.345865	105,764.411028	0.000000
4	35.294118	47.058824	53,834.698436	0.005612
5	34.117647	58.823529	91,048.593350	0.004488
6	35.338346	46.616541	76,300.125313	0.000000
7	28.007519	119.924812	148,796.992481	0.000000
8	40.000000	0.000000	77,372.781065	0.000000
9	32.941176	70.588235	42,352.941176	0.012706
10	30.075188	99.248120	57,636.949517	0.000000
11	20.000000	200.000000	90,000.000000	0.032000
12	34.705882	52.941176	72,710.432239	0.000515
13	30.451128	95.488722	71,478.696742	0.000000
14	33.529412	64.705882	59,911.921988	0.008508
15	40.000000	0.000000	0.000000	0.000000

- Store 1 has a  $\Delta x_1$  value of 30.075188 and a  $\Delta x_2$  value of 99.248120, with  $\Delta y_1$  at 54,135.34 and  $\Delta y_2$  at 0.00. This indicates that the store can potentially achieve an increase in input efficiency while maintaining its current output levels.
- Store 2 has a higher  $\Delta x_1$  value of 34.586466 and a lower  $\Delta x_2$  value of 54.135338 compared to Store 1. This store achieves a  $\Delta y_1$  value of 93,794.73, which suggests that it can allocate its budget more efficiently.
- Interestingly, Store 15 shows a  $\Delta x_1$  value of 40.000000 and a  $\Delta x_2$  value of 0.000000, while both  $\Delta y_1$  and  $\Delta y_2$  are equal to 0.00. This indicates that there are no improvements in the input efficiencies for this store, suggesting that its current allocation of resources is optimal given the budget constraint of 80,000.

Now we observe the behaviour of store 13 with different Budget levels as shown in below table; In examining the performance of Store 13 under various budget levels,

Table 3.6: Store 13 under different budget levels

Budget	$\Delta x_1$ (Change in Store Area)	$\Delta x_2$ (Change in Man-hours)	$\Delta y_1$ (Change in Revenue (INR))	$\Delta y_2$ (Change in Customer Loyalty (0-1))
80,000	30.451128	95.488722	71,478.70	0.000000
150,000	57.058824	179.411765	117,647.06	0.001294
200,000	77.647059	223.529412	144,117.65	0.009235
400,000	160.000000	400.000000	250,000.00	0.041000
600,000	242.352941	576.470588	355,882.35	0.072765
800,000	324.705882	752.941176	461,764.71	0.104529

we observe a notable relationship between the budget allocation and the changes in input and output efficiencies.

As the budget increases from INR 80,000 to INR 800,000, we see consistent growth in both  $\Delta x_1$  and  $\Delta x_2$  values. This suggests that the store can allocate its resources

more effectively with larger budgets, improving input efficiency. For example, when the budget is INR 80,000, the  $\Delta x_1$  value is 30.451128 and the  $\Delta x_2$  value is 95.488722. However, when the budget reaches INR 800,000, the  $\Delta x_1$  value rises to 324.705882 and the  $\Delta x_2$  value increases to 752.941176.

In terms of output efficiency, the  $\Delta y_1$  values show a steady increase as the budget grows, indicating that the store can generate higher outputs with a larger budget. Moreover, the  $\Delta y_2$  values remain relatively low, suggesting that the store can maintain its current output levels even when the budget is increased.

In summary, the analysis of Store 13's performance using income and budget-based models offers valuable insights for efficient resource allocation and revenue growth. These models help identify adjustments in store area, man-hours, and customer loyalty to balance financial and non-financial performance indicators. By combining the insights, decision-makers can optimize store operations, effectively allocate resources, and contribute to the retail chain's overall success. Continual assessment and adaptation of strategies based on these models will ensure long-term sustainability and growth in a competitive market.

# Chapter 4

## Stochastic DEA

### Introduction

Measuring the efficiency of organizations, especially non-profit ones that depend on public funds, has become an important research area in recent years. Various methodological approaches, such as the deterministic frontier, stochastic frontier, and mathematical programming approaches, have been employed to measure efficiency in this context. The mathematical programming approach is fundamentally non-parametric and has several benefits that make it attractive for efficiency measurement [4].

Data Envelopment Analysis (DEA) is a non-parametric frontier estimation methodology based on linear programming that measures the relative efficiencies and performance of Decision Making Units (DMUs) that transform multiple inputs into multiple outputs [2]. DEA is used to classify DMUs as either efficient or inefficient and to identify the source and level of inefficiency. However, most previous DEA applications have considered input variables of deterministic nature, although some may have a random nature. Therefore, stochastic DEA models have been proposed to handle random input variables [14].

In this Chapter, we propose a modification to the existing stochastic DEA model by introducing a chance-constrained input-oriented DEA model and also we will assume the inputs and outputs are random variables and present an uncertain DEA model and its equivalent crisp model. The proposed model is illustrated with a hypothetical example, and a solving algorithm is also presented. The rest of the paper is organized as follows. Section 2 discusses the methodology of the general DEA model and stochastic DEA model. Section 3 includes the proposed stochastic chance-constrained input-oriented DEA model, the proposed solving algorithm, and an illustrative example. Finally, the paper concludes with implications for the future.

This study aims to contribute to the existing literature on efficiency measurement in non-profit organizations by proposing a modified stochastic DEA model that considers random input variables. The proposed model can help decision-makers identify the source and level of inefficiency.

## 4.1 Developing an input-oriented DEA model [11]

The classical DEA model is based on maximizing a production function that represents a deterministic frontier, where the DEA estimate defines the maximum output producible from inputs under all circumstances, and the minimum input producing a given output under all circumstances. This method is comparable to a one-sided deviation estimated using mathematical programming methods.

The DEA model can be classified as either having CRS or VRS (0.1), depending on the assumptions relating to changes in outputs resulting from changes in inputs. Under the CRS model, the outputs change in direct proportion to the change in inputs, assuming that the scale of operation does not influence efficiency. Thus, the output and input-oriented measures of efficiency are equal. However, under the VRS model, changes in outputs are not necessarily proportional to the changes in inputs, meaning that the output and input-oriented measures of efficiency scores are not equal for inefficient units. In this section, we are specifically focused on the input-oriented VRS model (2.1).

The standard DEA model assumes a deterministic production function, but Subhash Ray proposed a modification to the model that can account for random variation in all outputs produced from given inputs. The DEA estimate still defines the maximum output producible from inputs under all circumstances, but in this stochastic output-oriented model, the inputs are assumed to be deterministic while all outputs are random. Each output is normally distributed with a mean and variance, and the relation between the same stochastic output variable through different DMUs is independent ( $cov(y_k, y_p) = 0$ ), meaning that the random inequality in the DEA model may sometimes be violated.

To account for this, the chance-constrained programming (CCP) approach is used to ensure that the probability that the inequality holds for a random sample of these variables does not fall below a certain level. The chance-constrained output-oriented

model and input-oriented model to measure the efficiency level of a  $DMU_p$  are given in the Appendix section(Data) table(C.2) , I am going to explain the construction of the Input-Oriented stochastic model and similar explanations will work for Output-oriented stochastic model.

## Developing Input-Oriented Stochastic DEA [11]

To evaluate the performance of public Higher Education Institutions (HEIs) and ensure quality, it is important to consider the possibility of stochastic nature in some of the input variables. As a result, a Stochastic Input Oriented DEA model has been developed to modify the standard DEA model and measure technical efficiency in the presence of random variation in some of the inputs. This model is based on the CCP method and addresses the random inequality that may be violated in the restriction involving some of the input quantities. CCP ensures that the probability that the inequality holds for a random sample of these variables does not fall below a certain level. The chance-constrained input-oriented model is used to apply this strategy and provide measurements of technical efficiency.

The following is the chance-constrained input-oriented model used to measure the efficiency level of  $DMU_p$  [11]:

$$\begin{aligned}
 & \text{Min } Z_p = \theta \\
 & \text{Subject to} \\
 & pr \left\{ \sum_{i=1}^n \lambda_i x_i \leq \theta x_p \right\} \geq (1 - \alpha_j), \quad \forall j = 1, 2, \dots, J_s \\
 & \sum_{i=1}^n \lambda_i x_i \leq \theta x_p, \quad \forall j = 1, \dots, J_D \\
 & \sum_{i=1}^n \lambda_i y_i \geq y_p, \quad (\forall k = \{1, \dots, s\}) \\
 & \sum_{i=1}^n \lambda_i = 1 \\
 & \lambda_i \geq 0, \quad i = 1, 2, \dots, n
 \end{aligned} \tag{4.1}$$

Where  $n$  = number of DMus,  $m$  = the number of inputs  $s$  = the number of outputs,  $J_s$  = number of Stochastic inputs,  $J_D$  = number of deterministic inputs.

The inputs can be split into two categories: some are fixed and some are subject to random variation. The random inputs  $x_j, j \in J_s$  are assumed to follow a normal distribution, with a certain average value  $\mu_j$  and a level of variability  $\sigma_j^2$ . Additionally, the relationship between these stochastic inputs across different DMUs is not independent i.e.  $covariance(x_i, x_p) \neq 0$ , meaning that they are somehow related or dependent on each other.

Let  $u$  be a random variable defined as

$$u = \sum_{i=1}^n \lambda_i x_i - \theta x_p \leq 0, \quad \text{where } \lambda_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.2)$$

Taking the mean of  $u$ , we get

$$\mu_u = E(u) = E\left(\sum_{i=1}^n \lambda_i x_i - \theta x_p\right) = \sum_{i=1}^n \lambda_i E(x_i) - \theta E(x_p) = \sum_{i=1}^n \lambda_i \mu_i - \theta \mu_p \quad (4.3)$$

The variance of  $u$  is given by

$$\sigma_u^2 = var(u) = \sum_{i=1, i \neq p}^n \lambda_i^2 \sigma_i^2 + (\lambda_p - \theta)^2 \sigma_p^2 + 2cov(x_i, x_p) \quad (4.4)$$

Given that the variables  $x_j$  are normally distributed, their linear combination  $u$  is also normally distributed with mean  $\mu_u$  and variance  $\sigma_u^2$ . Thus, we can calculate the probability of  $u \leq 0$  using the standard normal distribution by transforming  $u$  into its standardized normal equivalent value  $z$ . Using the CDF of the standard normal distribution  $\Phi(\cdot)$ , we have:

$$pr \left\{ z \leq \frac{-\mu_u}{\sigma_u} \right\} = 1 - \Phi \left( \frac{\mu_u}{\sigma_u} \right) \quad (4.5)$$

We can replace the random inequality restriction in the Chance Constrained DEA Problem (CCDEAP) (4.1) with an equivalent restriction, which is given by  $1 - \Phi \left( \frac{\mu_u}{\sigma_u} \right) \geq 1 - \alpha \iff \Phi \left( \frac{\mu_u}{\sigma_u} \right) \leq \alpha = \Phi(e)$ , where  $\alpha$  is the confidence level, and  $e$  is the standard normal quantile corresponding to the desired level of confidence.

Since  $\Phi(\cdot)$  is an increasing function, we can simplify this restriction to  $\mu_u \leq e\sigma_u$ , substituting equation (4.3) and (4.4) in  $\mu_u \leq e\sigma_u$ , we get following equivalent deterministic model of (4.1) ;

$$\begin{aligned}
& \text{Min } Z_p = \theta \\
& \text{Subject to} \\
& \sum_{i=1}^n \lambda_i \mu_i - \theta \mu_p \leq e \sqrt{\sum_{i=1, i \neq p}^n \lambda_i^2 \sigma_i^2 + (\lambda_p - \theta)^2 \sigma_p^2 + 2cov(x_i, x_p)} \quad \forall j = 1, 2, \dots, J_s \\
& \sum_{i=1}^n \lambda_i x_i \leq \theta x_p, \quad \forall j = 1, \dots, J_D \\
& \sum_{i=1}^n \lambda_i y_i \geq y_p, \quad (\forall k = \{1, \dots, s\}) \\
& \sum_{i=1}^n \lambda_i = 1 \\
& \lambda_i \geq 0, \quad i = 1, 2, \dots, n
\end{aligned} \tag{4.6}$$

The Non-Linear programming model that was developed above can be employed to assess the efficiency of different DMUs in relation to each other.

### Illustration with example

To demonstrate the practical use of this model, we have taken the hypothetical data for three universities (11). By using this model, we can compare the relative efficiency of each university with respect to the others.

Table (B.1) presents the data for the deterministic input and output variables of three hypothetical universities. The input variables are the number of professors in each university, and the output variables are the number of diplomas, bachelor's, and master's granted in a year by each university. The table shows the values for these variables for each university.

Table (B.1) provides the parameters of the stochastic input variable which is the annual budget of each university (DMU). The data in the table represent the mean and variance of the budget for each university. The budget variable is assumed to be normally distributed with different mean and variance for each university.

In this problem, we are assuming a significance level of 5%, which corresponds to a value of 1.96. As explained in Model (16), we need to provide a Non-Linear Programming (NLP) formulation for each university in order to measure their relative efficiency. For evaluating the relative efficiency of University A, we need to solve a specific NLP problem(11).

Minimize  $Z_A = \theta$  s.t.

$$\begin{aligned}
 14\lambda_A + 15\lambda_B + 12\lambda_C - 14\theta &\leq 1.96 \left[ \sqrt{1.5\lambda_B^2 + 1.4(\lambda_A - \theta)^2 + 2 \cdot 0.9} \right. \\
 &\quad \left. + \sqrt{1.2\lambda_C^2 + 1.4(\lambda_A - \theta)^2 + 2 \cdot 0.6} \right] \\
 5\lambda_A + 8\lambda_B + 7\lambda_C &\leq 5\theta \\
 9\lambda_A + 5\lambda_B + 4\lambda_C &\geq 9 \\
 4\lambda_A + 7\lambda_B + 9\lambda_C &\geq 4 \\
 16\lambda_A + 10\lambda_B + 13\lambda_C &\geq 16 \\
 \lambda_A + \lambda_B + \lambda_C &= 1 \\
 \lambda_A, \lambda_B, \lambda_C &\geq 0
 \end{aligned}$$

To obtain the relative efficiency of university A, we can solve the problem using a suitable library in Python. The code for this problem can be found in the appendix chapter(A.3) and the optimal value obtained is the Optimal Solution, similarly we can adjust the coefficient of appropriate data to calculate the efficiency of the university B and C. below is the table for the efficiency. The values of  $\lambda_A$ ,  $\lambda_B$ ,  $\lambda_C$ , Theta, and Efficiency have been rounded to two decimal places where applicable, and scientific notation has been used for small values.

	Lambda_A	Lambda_B	Lambda_C	Theta	% Efficiency
University A	1.0	0.0	2.22e-16	1.00	100.00(efficient)
University B	0.40	0.0	0.60	0.78	77.50(inefficient)
University C	4.99e-16	0.0	1.0	1.00	100.00(efficient)

Table 4.1: Optimal solution table



The above analysis indicates that Universities A and C demonstrate efficiency in their performance, while University B falls short in this regard. Essentially, this means that University B produces lower outputs despite having a considerable amount of resources at its disposal, specifically a larger number of professors and a higher budget allocation compared to the other two institutions. As we see in the table (B.1) and (B.1) that University B has the greatest input levels among the three universities but fails to achieve proportionately high outputs, thereby reflecting its inefficiency which is verified by above model.

In conclusion, the model (4.6) presents a promising approach for evaluating performance in the context of multiple inputs and outputs, especially when dealing with a mix of deterministic and stochastic variables. Future work will focus on applying the (4.6) model to real-world cases and further expanding the model to accommodate various probability distributions and incorporate both stochastic input and output variables.

## 4.2 Linearization of Stochastic DEA through Symmetric Error Structure(3)

In the previous section, we developed a non-linear stochastic input-oriented model based on Chance Constrained Programming, which can be challenging to solve in certain cases. In this section, we will introduce a symmetric error structure that allows us to obtain an equivalent non-linear deterministic model for the stochastic CCR model. Subsequently, we will transform this deterministic model into a linear form, thereby simplifying the solution process.

Let We have a set of  $n$  DMUs, each with  $m$  inputs denoted by  $x_{ij}$  and  $s$  outputs represented by  $y_{rj}$ , as specified in the Notations page(6). All of these input and output vectors are non-negative and non-zero. The production possibility set for the CCR model, defined as  $T_{CCR} = T^{(CRS)}$ , is given in chapter 2 (2.2). In order to assess the efficiency of the target  $DMU_o$ , we employ the input-oriented and output-oriented radial form of the CCR model, as presented in the table (2.6,2.11) as below:

In the aforementioned CCR model, input and output vectors are considered to have deterministic values. However, let's examine a scenario in which these data points are treated as random variables. We will assume that random input and output vectors are provided, along with their respective expectation vectors. Furthermore,

	<b>Input-oriented CCR</b>	<b>Output-oriented CCR</b>
<b>Objective Function</b>	$\theta_o(\alpha) = \min \theta$	$\varphi_o(\alpha) = \max \varphi$
<b>Constraint 1</b>	$\sum_{j=1}^n x_{ij} \lambda_j \leq \theta x_{io}, i = 1, \dots, m$	$\sum_{j=1}^n x_{ij} \lambda_j \leq x_{io}, i = 1, \dots, m$
<b>Constraint 2</b>	$\sum_{j=1}^n y_{rj} \lambda_j \geq y_{ro}, r = 1, \dots, s$	$\sum_{j=1}^n y_{rj} \lambda_j \geq \varphi y_{ro}, r = 1, \dots, s$
<b>Constraint 3</b>	$\lambda_j \geq 0, j = 1, \dots, n$	$\mu_j \geq 0, j = 1, \dots, n$

Table 4.2: input-oriented and output-oriented CCR radial-form efficiency models

each input and output is assumed to follow a normal distribution, as described in the table below:

	<b>Random Vector</b>	<b>Expectations</b>	<b>Distributions</b>
Input	$\tilde{X}_j = (\tilde{x}_{1j}, \dots, \tilde{x}_{mj})$	$X_j = (x_{1j}, \dots, x_{mj})$	$\tilde{x}_{ij} \sim N(x_{ij}, \sigma_{ij}^2)$
Output	$\tilde{Y}_j = (\tilde{y}_{1j}, \dots, \tilde{y}_{sj})$	$Y_j = (y_{1j}, \dots, y_{sj})$	$\tilde{y}_{rj} \sim N(y_{rj}, \sigma_{rj}^2)$

Table 4.3: Vectors, random vectors, expected values, and distributions

Thus, the chance-constrained model associated with the input-oriented and output-oriented stochastic CCR model for evaluating  $DMU_o$ , where  $o$  belongs to the set  $\{1, \dots, n\}$ , in the level of significance  $\alpha (\alpha \in [0, 1])$  is as follows:

	<b>Input-oriented CCR</b>	<b>Output-oriented CCR</b>
<b>Objective Function</b>	$\theta_o(\alpha) = \min \theta$	$\varphi_o(\alpha) = \max \varphi$
<b>Constraint 1</b>	$P \left\{ \sum_{j \in \{1, \dots, n\}} \lambda_j \tilde{x}_{ij} \leq \theta \tilde{x}_{io} \right\} \geq 1 - \alpha$	$P \left\{ \sum_{j \in \{1, \dots, n\}} \mu_j \tilde{x}_{ij} \leq \tilde{x}_{io} \right\} \geq 1 - \alpha$
<b>Constraint 2</b>	$P \left\{ \sum_{j \in \{1, \dots, n\}} \lambda_j \tilde{y}_{rj} \geq \tilde{y}_{ro} \right\} \geq 1 - \alpha$	$P \left\{ \sum_{j \in \{1, \dots, n\}} \mu_j \tilde{y}_{rj} \geq \varphi \tilde{y}_{ro} \right\} \geq 1 - \alpha$
<b>Constraint 3</b>	$\lambda_j \geq 0; j = 1, \dots, n$	$\mu_j \geq 0; j = 1, \dots, n$

Table 4.4: Input and output-oriented chance constraint CCR SDEA

Let  $\Phi$  be a joint standard normal distribution function. Since  $\Phi$  is invertible ( $\Phi^{-1}(\alpha)$  represents the inverse of the cumulative distribution function (CDF) at the level of significance  $\alpha$ .) function, we can use this to convert the above chance-constrained model in the equivalent deterministic form which is a non-linear and quadratic programming problem.

The deterministic equivalent of the chance-constrained input-oriented CCR model, derived by Cooper et al. ([9]), and output-oriented CCR ([1]) is as follows:

**deterministic form of Input-oriented stochastic equivalent model:**

$$\begin{aligned}
& \theta_o(\alpha) = \min \theta \\
& \text{s.t.} \\
& \sum_{j=1}^n x_{ij} \lambda_j + s_i^- - \Phi^{-1}(\alpha) v_i = \theta x_{io}, \quad i = 1, \dots, m, \\
& \sum_{j=1}^n y_{rj} \lambda_j - s_r^+ + \Phi^{-1}(\alpha) u_r = y_{ro}, \quad r = 1, \dots, s, \\
& v_i^2 = \sum_{j \neq o} \sum_{k \neq o} \lambda_j \lambda_k \text{cov}(\tilde{x}_{ij}, \tilde{x}_{ik}) + 2(\lambda_o - \theta) \sum_{j \neq o} \lambda_j \text{cov}(\tilde{x}_{ij}, \tilde{x}_{io}) + (\lambda_o - \theta)^2 \text{var}(\tilde{x}_{io}), \quad i = 1, \dots, m, \\
& u_r^2 = \sum_{j \neq o} \sum_{k \neq o} \lambda_j \lambda_k \text{cov}(\tilde{y}_{rj}, \tilde{y}_{rk}) + 2(\lambda_o - 1) \sum_{j \neq o} \lambda_j \text{cov}(\tilde{y}_{rj}, \tilde{y}_{ro}) + (\lambda_o - 1)^2 \text{var}(\tilde{y}_{ro}), \quad r = 1, \dots, s, \\
& s_i^- \geq 0, \quad s_r^+ \geq 0, \quad i = 1, \dots, m, \quad r = 1, \dots, s, \\
& \lambda_j \geq 0, \quad u_r \geq 0, \quad v_i \geq 0, \quad j = 1, \dots, n, \quad r = 1, \dots, s, \quad i = 1, \dots, m.
\end{aligned} \tag{4.7}$$

**deterministic form of Output-oriented stochastic equivalent model:**

$$\begin{aligned}
& \varphi_o(\alpha) = \max \varphi \\
& \text{s.t.} \\
& \sum_{j=1}^n x_{ij} \mu_j + s_i^- - \Phi^{-1}(\alpha) v_i = x_{io}, \quad i = 1, \dots, m, \\
& \sum_{j=1}^n y_{rj} \mu_j - s_r^+ + \Phi^{-1}(\alpha) u_r = \varphi y_{ro}, \quad r = 1, \dots, s, \\
& u_r^2 = \sum_{j \neq o} \sum_{k \neq o} \mu_j \mu_k \text{cov}(\tilde{x}_{ij}, \tilde{x}_{ik}) + 2(\mu_o - \varphi) \sum_{j \neq o} \mu_j \text{cov}(\tilde{x}_{ij}, \tilde{x}_{io}) + (\mu_o - \varphi)^2 \text{var}(\tilde{x}_{io}), \quad r = 1, \dots, s, \\
& v_i^2 = \sum_{j \neq o} \sum_{k \neq o} \mu_j \mu_k \text{cov}(\tilde{y}_{rj}, \tilde{y}_{rk}) + 2(\mu_o - 1) \sum_{j \neq o} \lambda_j \text{cov}(\tilde{y}_{rj}, \tilde{y}_{ro}) + (\lambda_o - 1)^2 \text{var}(\tilde{y}_{ro}), \quad s = 1, \dots, m, \\
& s_i^- \geq 0, \quad s_r^+ \geq 0, \quad i = 1, \dots, m, \quad r = 1, \dots, s, \\
& \mu_j \geq 0, \quad u_r \geq 0, \quad v_i \geq 0, \quad j = 1, \dots, n, \quad r = 1, \dots, s, \quad i = 1, \dots, m.
\end{aligned} \tag{4.8}$$

**Definition 9.** *At the significance level  $\alpha$ , a DMU<sub>o</sub> is considered stochastically output-oriented (or input-oriented) weakly efficient if  $\varphi_o(\alpha) = 1$  (or  $\theta_o(\alpha) = 1$ , respectively).*

The following theorem is established by Cooper et al. (9)

**Theorem 5.** For models (4.7) and (4.8), the following properties hold:

1. The models are feasible across all levels of significance, denoted by  $\alpha$ .
2. When  $\alpha \leq 0.5$ , the conditions  $0 < \theta_o(\alpha) \leq 1$  and  $\varphi_o(\alpha) \geq 1$  are satisfied.
3. If  $\alpha \leq \alpha_0$ , then it is true that  $\theta_o(\alpha_0) \leq \theta_o(\alpha)$  and  $\varphi_o(\alpha_0) \geq \varphi_o(\alpha)$ .

### Stochastic Efficiency of DMUs based on Symmetric-error structure in inputs and outputs 3

Now we introduce a symmetric error structure for random inputs and outputs. By utilizing this structure, the stochastic CCR model can be transformed into a deterministic linear model.

Assume that the input and output levels of the Decision-Making Units (DMUs) exhibit a symmetric error structure. For each DMU  $j$ , where  $j \in \{1, 2, \dots, n\}$ , the inputs and outputs can be represented as follows:

$$\begin{aligned}\tilde{x}_{ij} &= x_{ij} + a_{ij}\tilde{\epsilon}_{ij}, & \forall i \in \{1, 2, \dots, m\}; \\ \tilde{y}_{rj} &= y_{rj} + b_{rj}\tilde{\xi}_{rj}, & \forall r \in \{1, 2, \dots, s\}.\end{aligned}$$

In this representation,  $a_{ij}$  and  $b_{rj}$  are non-negative real values. The variables  $\tilde{\epsilon}_{ij}$  and  $\tilde{\xi}_{rj}$  are normally distributed random variables with mean 0 and variance  $\bar{\sigma}^2$ . These variables represent the errors in the input and output levels compared to their respective mean values.

Given that the normal distribution is symmetric, this structure in the given expression is referred to as a symmetric error structure. Moreover, the following relationships can be derived from the expression:

$$\begin{aligned}\tilde{x}_{ij} &\sim N(x_{ij}; \bar{\sigma}^2 a_{ij}^2); \\ \tilde{y}_{rj} &\sim N(y_{rj}; \bar{\sigma}^2 b_{rj}^2).\end{aligned}$$

It is important to note that any random variable with a normal distribution can be represented as a symmetric error structure. Assume that the  $i$ th input of each DMU is uncorrelated. Similarly, assume the  $r$ th output of each DMU is also uncorrelated. That is, for any  $j \neq k$ ,

$$\text{Cov}(\tilde{\epsilon}_{ij}, \tilde{\epsilon}_{ik}) = 0; \quad i = 1, \dots, m; \quad \text{Cov}(\tilde{\xi}_{rj}, \tilde{\xi}_{rk}) = 0; \quad r = 1, \dots, s. \quad (4.9)$$

Taking into account the relationships described above, we can consider a common error for all DMUs, meaning  $\tilde{\epsilon}_i = \tilde{\epsilon}_{ij}$  and  $\tilde{\xi}_r = \tilde{\xi}_{rj}$ , for every  $j = 1, \dots, n$ ,  $i = 1, \dots, m$ , and  $r = 1, \dots, s$ . Now, we derive the linear programming form of the chance-constrained program according to M. H. Behzadi [3] as follows:

Let's consider the  $i$ -th input constraint1 of the input-oriented model as in table (4.4).

$$\mathbb{P} \left\{ \sum_{j=1}^n \tilde{x}_{ij} \lambda_j \leq \theta \tilde{x}_{io} \right\} \geq 1 - \alpha \quad (4.10)$$

suppose

$$\tilde{h}_i = \sum_{j=1}^n \lambda_j \tilde{x}_{ij} - \theta \tilde{x}_{io}$$

Then from (4.10) and (4.9), we get

$$\tilde{h}_i = \left( \sum_{j=1}^n \lambda_j x_{ij} - \theta x_{io} \right) + \tilde{\epsilon}_i \left( \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} \right). \quad (4.11)$$

Thus,  $\tilde{h}_i$  follows a normal distribution:

$$\tilde{h}_i \sim N \left( \sum_{j=1}^n \lambda_j x_{ij} - \theta x_{io}, \bar{\sigma}^2 \left( \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} \right)^2 \right). \quad (4.12)$$

Using the properties of normal distribution, we can convert the stochastic constraint (4.10) into a deterministic equivalent

For the  $i$ -th constraint (4.10), the following deterministic equivalent can be derived:

$$\sum_{j=1}^n \lambda_j x_{ij} - \phi^{-1}(\alpha) \bar{\sigma} \left| \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} \right| \leq \theta x_{io}; \quad i = 1, \dots, m. \quad (4.13)$$

Similarly, the  $r$ -th output constraint (constraint2 in table (4.4)) can be converted as:

$$\sum_{j=1}^n \lambda_j y_{rj} + \phi^{-1}(\alpha) \bar{\sigma} \left| \sum_{j=1}^n \lambda_j b_{rj} - b_{ro} \right| \geq y_{ro}; \quad r = 1, \dots, s. \quad (4.14)$$

Thus, the deterministic equivalent model of input-oriented CCR as in table (4.4) can be represented as:

$$\begin{aligned}
& \min \theta \\
& \text{s.t.} \\
& \sum_{j=1}^n \lambda_j x_{ij} - \phi^{-1}(\alpha) \bar{\sigma} \left| \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} \right| \leq \theta x_{io}; \quad i = 1, \dots, m, \\
& \sum_{j=1}^n \lambda_j y_{rj} + \phi^{-1}(\alpha) \bar{\sigma} \left| \sum_{j=1}^n \lambda_j b_{rj} - b_{ro} \right| \geq y_{ro}; \quad r = 1, \dots, s, \\
& \lambda_j \geq 0; \quad j = 1, \dots, n.
\end{aligned}$$

now we convert this non-linear model to its equivalent form by using the below information

$$\begin{aligned}
|x| &= \max\{0, x\} + \max\{0, -x\} = u + v, \\
&\text{where } u = \max\{0, x\} \quad \text{and} \quad v = \max\{0, -x\}, \\
&x = u - v,
\end{aligned}$$

with the condition  $u, v \geq 0$  and  $uv = 0$ .

by applying this result and introducing the notations  $p_i^+, p_i^-$  and  $q_r^+, q_r^-$ , we get the following conditions [3],

$$\begin{aligned}
& \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} = (p_i^+ + p_i^-); \quad i = 1, \dots, m; \\
& \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} = (p_i^+ - p_i^-); \quad i = 1, \dots, m; \\
& p_i^+ p_i^- = 0; \quad i = 1, \dots, m; \\
& \sum_{j=1}^n \lambda_j b_{rj} - b_{ro} = (q_r^+ + q_r^-); \quad r = 1, \dots, s; \\
& \sum_{j=1}^n \lambda_j b_{rj} - b_{ro} = (q_r^+ - q_r^-); \quad r = 1, \dots, s; \\
& q_r^+ q_r^- = 0; \quad r = 1, \dots, s; \\
& p_i^+ \geq 0; \quad p_i^- \geq 0; \quad q_r^+ \geq 0; \quad q_r^- \geq 0.
\end{aligned}$$

and by substituting these conditions into the above model we get the following non-linear program(Non-linearity because of constraints  $p_i^+ p_i^- = 0, q_r^+ q_r^- = 0$ )

$$\begin{aligned}
& \min \theta \quad \text{s.t.:} \\
& \sum_{j=1}^n \lambda_j x_{ij} - \Phi^{-1}(\alpha)(p_i^+ + p_i^-) \leq \theta x_{io}; \quad i = 1, \dots, m; \\
& \sum_{j=1}^n \lambda_j y_{rj} + \Phi^{-1}(\alpha)(q_r^+ + q_r^-) \geq y_{ro}; \quad r = 1, \dots, s; \\
& \sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} = p_i^+ - p_i^-; \quad i = 1, \dots, m; \\
& \sum_{j=1}^n \lambda_j b_{rj} - b_{ro} = q_r^+ - q_r^-; \quad r = 1, \dots, s; \\
& p_i^+ p_i^- = 0; \quad i = 1, \dots, m; \\
& q_r^+ q_r^- = 0; \quad r = 1, \dots, s; \\
& \lambda_j, p_i^+, p_i^-, q_r^+, q_r^- \geq 0; \quad j = 1, \dots, n; \quad i = 1, \dots, m; \quad r = 1, \dots, s.
\end{aligned}$$

This model is nonlinear due to constraints involving the product of decision variables. However, since any linear problem with an optimal solution has a basic optimal solution, these constraints are automatically satisfied in optimal basic solutions. Thus, these constraints can be removed by utilizing algorithms like the Simplex method, resulting in a linear deterministic model as shown in the table below (4.5) ( Similarly, we can obtain a linear deterministic form of a chance-constrained output-oriented model (4.4) as shown in below table (4.5))

As This Input-oriented linear model (4.5) is a special case of (4.7) this implies Theorem (5) is applicable for this special case also, and these theorem explanation for Input-oriented linear model derived as below:

Part 2 proof of theorem (3) establish that for the stochastic efficiency linear model, the optimal value of  $\theta$  lies in the range  $(0, 1]$  for every  $\alpha$  level less than 0.5. The proof demonstrates that the model is minimizing  $\theta$  and that  $\theta$  cannot be equal to or less than 0 if  $\alpha$  is less than 0.5.

Table 4.5: Stochastic efficiency linear models

<b>Input-oriented</b>	<b>Output-oriented</b>
$\theta_o(\alpha) = \min \theta$	$\varphi_o(\alpha) = \max \varphi$
$\sum_{j=1}^n \lambda_j x_{ij} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq \theta x_{io}$ $i = 1, 2 \dots m$	$\sum_{j=1}^n \mu_j x_{ij} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq x_{io}$ $i = 1, 2 \dots m$
$\sum_{j=1}^n \lambda_j y_{rj} + \phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq y_{ro}$ $r = 1, 2 \dots s$	$\sum_{j=1}^n \mu_j y_{rj} + \phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq \varphi y_{ro}$ $r = 1, 2 \dots s$
$\sum_{j=1}^n \lambda_j a_{ij} - \theta a_{io} = p_i^+ - p_i^-$ $i = 1, 2 \dots m$	$\sum_{j=1}^n \mu_j a_{ij} - a_{io} = p_i^+ - p_i^-$ $i = 1, 2 \dots m$
$\sum_{j=1}^n \lambda_j b_{rj} - b_{ro} = q_r^+ - q_r^-$ $r = 1, 2 \dots s$	$\sum_{j=1}^n \mu_j b_{rj} - \varphi b_{ro} = q_r^+ - q_r^-$ $r = 1, 2 \dots s$
$\lambda \in \mathbb{R}^+, p_i^+ \geq 0, p_i^- \geq 0, q_r^+ \geq 0, q_r^- \geq 0$	$\mu \in \mathbb{R}^+, p_i^+ \geq 0, p_i^- \geq 0, q_r^+ \geq 0, q_r^- \geq 0$

part 3 and its proof [3] demonstrate that the optimal value of  $\theta$  decreases as the  $\alpha$  level increases, i.e.,  $\theta^*(\alpha') \leq \theta^*(\alpha)$  for every  $\alpha \leq \alpha'$  in deterministic linear model . This is shown by proving that an optimal solution of the model for a given  $\alpha$  level can also serve as a feasible solution for the model at a higher  $\alpha'$  level, due to the nondecreasing nature of the inverse of the standard normal distribution function,  $\Phi^{-1}$ .

Some implications of these two part of theorems are that if a decision-making unit (DMU) is efficient at  $\alpha'$  level of error, it will also be efficient for every  $\alpha$  level less than  $\alpha'$ . Conversely, if a DMU is inefficient at  $\alpha'$  level of error, it will remain inefficient for every  $\alpha$  level greater than  $\alpha'$ .

For  $\alpha = 0.5$ , the model (4.5) becomes the CCR model with mean data. This implies that if a DMU is efficient in the CCR mean model, it will be efficient for every  $\alpha$  level less than 0.5. In other words, permanently efficient DMUs can be identified by assessing them at  $\alpha = 0.5$ . Furthermore, if a DMU is inefficient at  $\alpha = 0$ , it will remain inefficient at all  $\alpha$  levels. By assessing DMUs at  $\alpha = 0.001$ , almost all permanently inefficient DMUs can be detected, since  $\Phi^{-1}(0) = -\infty$ , while  $\Phi^{-1}(0.001) \approx -3$  [3].



## An Application in Banking Sector

In this example, 20 branches of an Iranian bank are considered, each with three stochastic inputs and five stochastic outputs, as mentioned below: Inputs:

- Personal rate (weighted combination of personal qualifications, quantity, education, and others)
- Payable benefits (of all deposits)
- Delayed requisitions (delay in returning ceded loans and other facilities)

Outputs:

- Facilities (sum of business and individual loans)
- Amount of deposits (of current, short duration, and long duration accounts)
- Received benefits (of all ceded loans and facilities)
- Received commission (on banking operations, issuance guaranty, transferring money, and others)
- Other resources of deposits

The data, based on ten successive months, are assumed to have a normal distribution. The scaled parameters of the data are presented in appendix B as a table (B.3) and (B.4). The goal is to assess the total performance of these bank branches. In this example, it is assumed that the symmetric error structure has a value of  $\sigma = 1$ . Therefore, the parameters  $a_{ij}$  and  $b_{rj}$  are calculated as the square root of the variance of the respective inputs and outputs ( $\tilde{x}_{ij}$  and  $\tilde{y}_{rj}$ ), i.e.,  $a_{ij} = \sqrt{\text{Var}(\tilde{x}_{ij})}$  and  $b_{rj} = \sqrt{\text{Var}(\tilde{y}_{rj})}$ .

Using the stochastic efficiency linear model (4.5), we evaluated the efficiencies of all 20 branches and compiled the results in the below table with different levels of significance. The results show that the efficiency of each DMU increases as the level

of error decreases. Among the DMUs, 2, 3, 8, 9, 10, 11, 15, 16, and 19 are efficient at a level of error of 0.5 and are therefore permanent efficient DMUs. On the other hand, DMUs 1, 4, 5, 7, 12, 13, 14, 18, and 20 are inefficient at a level of error of 0.001 and are therefore permanent inefficient DMUs [3].

The efficiency calculations were performed using Python code, which is detailed in the Appendix A section (A.4). In this chapter, the analysis showed that it is important to

DMU	0.999	0.500	0.200	0.010	0.050	0.100	0.001
DMU1	0.726059	0.587251	0.552503	0.490487	0.519160	0.534287	0.457730
DMU2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU3	1.000000	1.000000	1.000000	0.988625	1.000000	1.000000	0.944770
DMU4	0.358747	0.276575	0.264015	0.241387	0.252158	0.257632	0.228499
DMU5	0.574451	0.487551	0.458993	0.397381	0.427294	0.442130	0.360809
DMU6	0.935613	0.919609	0.907225	0.880459	0.893233	0.899749	0.865694
DMU7	0.350661	0.480603	0.499988	0.514851	0.508678	0.504929	0.520333
DMU8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU10	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU11	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU12	0.654844	0.807305	0.847152	0.893232	0.872020	0.860829	0.917469
DMU13	0.701129	0.908487	0.930026	0.961053	0.947660	0.939972	0.974829
DMU14	0.750727	0.693021	0.660254	0.594807	0.623249	0.639464	0.565450
DMU15	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU16	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU17	0.775281	0.931433	0.953502	0.987924	0.972822	0.964291	1.000000
DMU18	0.620597	0.563222	0.544975	0.510523	0.526795	0.535105	0.491056
DMU19	0.980089	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DMU20	0.860498	0.475659	0.429495	0.361525	0.390521	0.407652	0.340587

Table 4.6: Efficiency results for each DMU with different significance levels

consider the level of error when evaluating stochastic efficiency using the equivalent nonlinear deterministic model of the stochastic CCR model with symmetric error structure. The study revealed that efficiency is a non-increasing function of the level of error, and a DMU may be permanently inefficient if it is not efficient at the lowest level of error. Future research should explore the application of symmetric error structures in other DEA models.

# Chapter 5

## Inverse DEA with Stochastic data

Efficiency evaluation is a fundamental issue in various fields, including business, economics, and management. The conventional data envelopment analysis (DEA) models assume deterministic data, but in many practical applications, we encounter stochastic data due to various factors, such as financial crises, social and political features, inflation, and natural components. To address this issue, stochastic DEA (SDEA) has been introduced, which utilizes production economic principles and statistical methods to estimate stochastic inefficiency rather than deterministic or stochastic production possibility sets (PPS).

One area of SDEA is inverse DEA (InvDEA), where an efficiency target is set as a strategic objective, and the primary goal is to determine the necessary input and output vectors to achieve this target. Previous research in InvDEA focused on solving specific problems using deterministic data. However, in this paper titled "Inverse Data Envelopment Analysis with Stochastic Data," Ali Ghomi et al. propose models for input/output estimation in the presence of stochastic data in the context of InvDEA.

To contribute to this area, we will provide Python code for these models, making it more accessible and applicable for researchers and practitioners in various fields. By implementing these models, users can estimate input/output levels in the presence of stochastic data and make more informed decisions regarding efficiency targets, resource allocation, and investment analysis.

## Some Information of Data

We use the notation of section 2 of Chapter 4 and Models of Chapter 3. We are analyzing a collection of Decision-Making Units (DMUs) where the input and output quantities follow normal probability distributions. Specifically, input levels  $\bar{x}_{ij}$  adhere to a normal distribution with mean  $x_{ij}$  and variance  $\sigma_{ij}^2$ , while output levels  $\bar{y}_{rj}$  are associated with a normal distribution that has a mean of  $y_{rj}$  and variance  $\psi_{rj}^2$ . Our objective is to estimate these parameters, which can be challenging to determine in real-world scenarios.

To accomplish this, we employ unbiased and efficient estimators. According to the stochastic Data Envelopment Analysis (DEA) literature, we gather a random sample of size  $p$  for every DMU $_j$ . The sample comprises input and output data labeled as  $\hat{x}_{ij}^t$  and  $\hat{y}_j^{tr}$ , respectively, where  $i$  is in the set  $I = 1, 2..m$ ,  $r$  is in the set  $O = 1, 2.., s$ , and  $t$  ranges from 1 to  $p$ .

Using the random samples, we can calculate unbiased estimators for the input and output means and variances as follows:

1. To calculate the mean of inputs ( $\bar{x}_{ij}$ ), average the random sample:

$$\bar{x}_{ij} = \frac{1}{p} \sum_{t=1}^p \hat{x}_{ij}^t$$

2. To calculate the variance of inputs ( $\bar{\sigma}_{ij}^2$ ), average the squared differences between the random sample values and the estimated mean:

$$\bar{\sigma}_{ij}^2 = \frac{1}{p-1} \sum_{t=1}^p (\hat{x}_{ij}^t - \bar{x}_{ij})^2$$

3. To calculate the mean of outputs ( $\bar{y}_{rj}$ ), average the random sample:

$$\bar{y}_{rj} = \frac{1}{p} \sum_{t=1}^p \hat{y}_j^{tr}$$

4. To calculate the variance of outputs ( $\bar{\psi}_{rj}^2$ ), average the squared differences between

the random sample values and the estimated mean:

$$\bar{\psi}_{rj}^2 = \frac{1}{p-1} \sum_{t=1}^p (\hat{y}_j^{tr} - \bar{y}_{rj})^2$$

These estimators are considered unbiased for determining the true mean and variance of inputs and outputs. Additionally, if the sample size is sufficiently large, then  $\bar{x}_{ij}$ ,  $\bar{y}_{rj}$ ,  $\bar{\sigma}_{ij}^2$ , and  $\bar{\psi}_{rj}^2$  are considered the optimal estimators for the mean and variance of inputs and outputs, respectively.

In conclusion, by utilizing random samples and unbiased estimators, we can estimate the mean and variance of input and output quantities for each DMU, which is essential for conducting further efficiency evaluations when dealing with stochastic data.

## 5.1 Inverse DEA in the presence of stochastic data

In traditional Data Envelopment Analysis (DEA) models, a Decision-Making Unit's (DMU) efficiency score is typically computed based on input and output levels. However, in Inverse DEA (InvDEA), the focus shifts to setting an efficiency target as a strategic objective. The primary goal of InvDEA is to identify the required input and output vectors to achieve the desired efficiency target. Q.L Wei [\[19\]](#) explored the critical question in the context of InvDEA:

**Question:** If the efficiency index of DMU<sub>o</sub> remains unchanged concerning other DMUs but the input/output levels increase, to what extent should the output/input levels of DMU<sub>o</sub> increase?

This question has been tackled using Multi-Objective Programming (MOP) problems in the literature. The authors also investigated several theoretical and practical aspects of InvDEA, including:

- Setting revenue targets

- Investment analysis
- Resource allocation
- Inter-temporal dependence of data
- Fuzzy data

However, the models developed in these works, including those by Q.L Wei [19], could not address input and output estimation in the presence of stochastic data. Consequently, this thesis attempts to estimate input and output levels in the presence of stochastic data, building upon the work of Q.L Wei [19].

### 5.1.1 Input level estimation

Considering the question posed in the first scenario of Classical DEA Models In Chapter 4. Then considering the input-oriented case by increasing the output from  $\tilde{y}_o$  to  $\tilde{\beta}_o = (\tilde{y}_o + \Delta\tilde{y}_o)$  with  $\Delta\tilde{y}_o \geq 0, \Delta\tilde{y}_o \neq 0$  then need to find input level  $\tilde{\alpha}_o = (\tilde{x}_o + \Delta\tilde{x}_o)$ . introduce new  $DMU_{n+1}$  after the change in outputs, then the below model will calculate the efficiency index of  $DMU_{n+1}$  in the level of significance  $\alpha$ ;

$$\begin{aligned} \theta_{n+1}(\alpha) &= \min \theta & (5.1) \\ \text{s.t. } P \left\{ \sum_{j=1}^n \lambda_j \tilde{x}_j + \lambda_{n+1} \tilde{\alpha}_o \leq \theta \tilde{\alpha}_o \right\} &\geq 1 - \alpha; \\ P \left\{ \sum_{j=1}^n \lambda_j \tilde{y}_j + \lambda_{n+1} \tilde{\beta}_o \geq \tilde{\beta}_o \right\} &\geq 1 - \alpha; \\ \delta_1 \left( \sum_{j=1}^n \lambda_j + \lambda_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) &= \delta_1, \\ \nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i &\in \{0, 1\}; i = 1, 2, 3 \end{aligned}$$

If optimal values of (5.1) and (4.4) remain same in level of significance  $\alpha$  (that is  $\theta_{n+1}(\alpha) = \theta_o(\alpha)$ ), then we say efficiency index remains unchanged. To find the input level, Ali Ghomi et al [13] proposed following Stochastic Multi-Objective programming (SMOP) problem in level of significance  $\alpha$ :

$$\begin{aligned} & \min(\tilde{\alpha}_{1o}, \tilde{\alpha}_{2o}, \dots, \tilde{\alpha}_{mo}) & (5.2) \\ \text{s.t. } & P \left\{ \sum_{j=1}^n \lambda_j \tilde{X}_j \leq \theta_o(\alpha) \tilde{\alpha}_o \right\} \geq 1 - \alpha; \\ & P \left\{ \sum_{j=1}^n \lambda_j \tilde{Y}_j \geq \tilde{\beta}_o \right\} \geq 1 - \alpha; \\ & P \left\{ \tilde{\alpha}_o \geq \tilde{Y}_o \right\} \geq 1 - \alpha; \\ & \delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\ & \nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3 \end{aligned}$$

Here  $(\lambda, \tilde{\alpha}_o)$  variables and  $\theta_o(\alpha)$  (Optimal value of input-oriented model in table (4.4) and  $\tilde{\beta}_o$  are constant. As we defined Weak-pareto solution, Ali Ghomi et al [13] defined Stochastic Pareto (SP) and Stochastic Weak-Pareto (SWP) solutions as follows;

**Definition 10. Stochastic Pareto (SP):** A SP solution of (5.2) at a given level of significance  $\alpha$  is a feasible solution  $\Delta = (\lambda^*, \tilde{\alpha}_{1o}^*, \dots, \tilde{\alpha}_{mo}^*)$  such that there is no other feasible solution  $\gamma = (\lambda, \tilde{\alpha}_{1o}, \dots, \tilde{\alpha}_{mo})$  with the following properties ;

$$P\{\tilde{\alpha}_{io} - \tilde{\alpha}_{io}^* \leq 0\} \geq 1 - \alpha \quad \text{for each } i = 1, 2, \dots, m;$$

$$P\{\tilde{\alpha}_{io} - \tilde{\alpha}_{io}^* \leq -\epsilon\} \geq 1 - \alpha \quad \text{for some } i = 1, 2, \dots, m;$$

Where  $\epsilon$  is a non-archimedean infinitesimal.

**Definition 11. Stochastic Weak Pareto (SWP):** A SP solution of (5.2) at a given level of significance  $\alpha$  is a feasible solution  $\Delta = (\lambda^*, \tilde{\alpha}_{1o}^*, \dots, \tilde{\alpha}_{mo}^*)$  such that there is no other feasible solution  $\gamma = (\lambda, \tilde{\alpha}_{1o}, \dots, \tilde{\alpha}_{mo})$  with the following properties ;

$$P\{\tilde{\alpha}_{io} - \tilde{\alpha}_{io}^* \leq -\epsilon\} \geq 1 - \alpha \quad \text{for some } i = 1, 2, \dots, m;$$

Ali Ghomi et al [13] established the following two theorems to answer the questions of input level estimation in the case of the stochastic framework.

**Theorem 6.** Assume that  $\Delta = (\lambda^*, \tilde{\alpha}_o^*)$  is an SP solution at the significance level  $\alpha$  for model (5.2), given that at least one of the following conditions is met:

1.  $P\{|\tilde{\alpha}_{oj}^* - \tilde{X}_o| < \varepsilon\} \geq 1 - \alpha$
2. There exists at least  $l \in I$  such that  $P\{\tilde{\alpha}_{lo}^* - \tilde{x}_{lo} \geq \varepsilon\} \geq 1 - \alpha$

where  $\varepsilon$  is a non-Archimedean infinitesimal. Then,

$$\theta_o(\alpha) = \theta_{n+1}(\alpha)$$

**Theorem 7.** Let  $\Pi = (\bar{\lambda}, \bar{\alpha}_o)$  be a feasible solution to SMOP (5.2) such that  $\theta_o(\alpha) = \theta_{n+1}(\alpha)$ . Then,  $\Pi$  is an SWP solution at the significance level  $\alpha$  for SMOP (5.2).

Now the question is When should we begin utilizing InvDEA? Let's assume that the output levels of  $DMU_o$  are raised from  $\tilde{Y}_o$  to  $\tilde{Y}_o + \Delta\tilde{Y}_o$ . We will take into account the following assumption at the significance level  $\alpha$ :

$$\begin{cases} H_0 : \theta_{n+1}(\alpha) = \theta_o(\alpha) \\ H_1 : \theta_{n+1}(\alpha) \neq \theta_o(\alpha) \end{cases} \quad \text{equivalently} \quad \begin{cases} H_0 : \tilde{\alpha}_o = \tilde{X}_o \\ H_1 : \tilde{\alpha}_o \neq \tilde{X}_o \end{cases}$$

As we have assumed

$$\tilde{x}_{ij} \sim N(x_{ij}, \sigma_{ij}^2), \forall i \in I = \{1, 2, \dots, m\} \implies \bar{x}_{ij} \sim N\left(x_{ij}, \frac{\sigma_{ij}^2}{p}\right)$$

Hence,  $\frac{\bar{x}_{ij} - x_{ij}}{\frac{\sigma_{ij}}{\sqrt{p}}}$  is t-distributed with degree of freedom  $p - 1$ . Then we have following  $(1 - \alpha)\%$  confidence interval for mean inputs  $x_{ij}$  ;

$$x_{ij}^l(\alpha) = \bar{x}_{ij} - t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\sigma}_{ij}}{\sqrt{p}} \quad x_{ij}^u(\alpha) = \bar{x}_{ij} + t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\sigma}_{ij}}{\sqrt{p}}$$



Similarly for mean outputs  $\beta_{rj}$  we have;

$$\beta_{rj}^l(\alpha) = \bar{\beta}_{rj} - t_{(\frac{\alpha}{2})} \frac{\bar{\psi}_{rj}}{\sqrt{p}} \quad \beta_{rj}^u(\alpha) = \bar{\beta}_{rj} + t_{(\frac{\alpha}{2})} \frac{\bar{\psi}_{rj}}{\sqrt{p}}$$

where l denotes the lower bound and u denotes the upper bound of the confidence interval.

Now assuming worst-case scenario(pessimistic) and best case scenario(Optimistic) we have lower and upper bounds for efficiency of  $DMU_{n+1}$  with  $(1 - \alpha)\%$  confidence interval [13](#):

$$\begin{aligned} \theta_{n+1}^l &= \min \theta \\ &\text{subject to;} \\ &\sum_{j \in J} \lambda_j x_{ij}^l(\alpha) + \lambda_{n+1} x_{in+1}^u(\alpha) \leq \theta x_{in+1}^u(\alpha), \quad \forall i \in I, \\ &\sum_{j \in J} \lambda_j \beta_{rj}^u(\alpha) + \lambda_{n+1} \beta_{rn+1}^l(\alpha) \geq \beta_{rn+1}^l(\alpha), \quad \forall r \in O, \\ &\delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\ &\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3 \end{aligned} \tag{5.3}$$

$$\begin{aligned} \theta_{n+1}^u &= \min \theta \\ &\text{subject to;} \\ &\sum_{j \in J} \lambda_j x_{ij}^u(\alpha) + \lambda_{n+1} x_{in+1}^l(\alpha) \leq \theta x_{in+1}^l(\alpha), \quad \forall i \in I, \\ &\sum_{j \in J} \lambda_j \beta_{rj}^l(\alpha) + \lambda_{n+1} \beta_{rn+1}^u(\alpha) \geq \beta_{rn+1}^u(\alpha), \quad \forall r \in O \\ &\delta_1 \left( \sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\ &\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3 \end{aligned}$$

The efficiency score, denoted as  $\varphi_{n+1}(\alpha)$ , lies within the range defined by its lower bound,  $\theta_{in+1}(\alpha)$ , and upper bound,  $\theta_{un+1}(\nu)$ . As a basis for comparison, Model [5.1](#) is employed, with the parameter  $\tilde{\alpha}_o$  set to  $\tilde{X}_o$ .

If the optimal value resulting from Model (5.1) does not fall within the aforementioned range, i.e.,  $[\theta_{l_{n+1}}(\alpha), \theta_{u_{n+1}}(\alpha)]$ , the method outlined in this subsection can be applied. This approach estimates the input levels in such a way that the efficiency score of  $DMU_o$  remains constant at the significance level  $\alpha$ , represented by  $\theta_o(\alpha)$ .

### 5.1.2 Output level estimation

Now we reverse the question we posed in input level estimation, stated in the second scenario of chapter 4. Let input level increased from  $\tilde{X}_o$  to  $\tilde{\alpha}_o = \tilde{X} + \Delta\tilde{X}$ ;  $\Delta\tilde{X} \geq 0, \Delta\tilde{X} \neq 0$ , need to estimate output levels  $\tilde{\beta}_o = (\tilde{y}_o + \Delta\tilde{y}_o)$  by keeping the efficiency  $\phi_o(\alpha)$  of  $DMU_o$  unchanged in level of significance  $\alpha$  (4.4).

$DMU_{n+1}$  : denotes new DMU after increasing the input and output levels of  $DMU_o$ ; Then the following model estimates the efficiency index of the new DMU in significance level of  $\alpha$ .

$$\begin{aligned}
\varphi_{n+1}(\alpha) &= \max \varphi \\
\text{s.t. } P \left\{ \sum_{j=1}^n \mu_j \tilde{X}_j + \mu_{n+1} \tilde{\alpha}_o \leq \tilde{\alpha}_o \right\} &\geq 1 - \alpha, \\
P \left\{ \sum_{j=1}^n \mu_j \tilde{Y}_j + \mu_{n+1} \tilde{\beta}_o \geq \varphi \tilde{\beta}_o \right\} &\geq 1 - \alpha, \\
\delta_1 \left( \sum_{j=1}^n \mu_j + \mu_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) &= \delta_1, \\
\nu \geq 0, \mu_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{5.4}$$

If optimal values of output oriented model (4.4) and (5.4) remains same then efficiency index  $\phi_o(\alpha) = \phi_{n+1}(\alpha)$ . Now we introduce the following SMOP model (13) for output estimation in the level of significance  $\alpha$  as below;

$$\begin{aligned}
& \max && \left( \tilde{\beta}_{1o}, \tilde{\beta}_{2o}, \dots, \tilde{\beta}_{so} \right) \\
& \text{s.t.} && P \left\{ \sum_{j \in J} \mu_j \tilde{X}_j \leq \tilde{\alpha}_o \right\} \geq 1 - \alpha, \\
& && P \left\{ \sum_{j \in J} \mu_j \tilde{Y}_j \geq \phi_o(\alpha) \tilde{\beta}_o \right\} \geq 1 - \alpha, \\
& && P \left\{ \tilde{\beta}_o \geq \tilde{Y}_o \right\} \geq 1 - \alpha, \\
& && \delta_1 \left( \sum_{j=1}^n \mu_j + \mu_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& && \nu \geq 0, \mu_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{5.5}$$

In the given problem,  $(\mu, \tilde{\beta}_o)$  represents the variable vector. The constant value  $\phi_o(\alpha)$  denotes the optimal value of the output-oriented model in Table (4.4). Furthermore,  $\tilde{\alpha}_o$  is a known and fixed value.

Following theorem established by Ali Ghomi et al (13) to answer the second scenario in stochastic case;

**Theorem 8.** *If  $\Delta = (\mu^*, \tilde{\beta}_o^* = (\tilde{\beta}_{1o}^*, \dots, \tilde{\beta}_{so}^*))$  is a SWP solution in the level of significance  $\alpha$  to model (5.5), then  $\phi_o(\alpha) = \phi_{n+1}(\alpha)$ .*

**Theorem 9.** *Let  $\Delta = (\mu, \tilde{\beta}_o = (\tilde{\beta}_{1o}^*, \tilde{\beta}_{2o}^*, \dots, \tilde{\beta}_{so}^*))$  be a feasible solution to SMOP (5.5) such that  $\phi_o(\alpha) = \phi_{n+1}(\alpha)$ . Then,  $\Delta$  is an SWP solution in the level of significance  $\alpha$  to SMOP (5.5).*

As we derived the optimistic and pessimistic view point in the previous section using statistical concept, similar concept will be applied to find the bound of efficiency of model (5.5) and we have the below result (5.5) ;

$$\begin{aligned}
\alpha_{ij}^l(\alpha) &= \bar{\alpha}_{ij} - t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\sigma}_{ij}}{\sqrt{p}}, & \alpha_{ij}^u(\alpha) &= \bar{\alpha}_{ij} - t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\sigma}_{ij}}{\sqrt{p}}, & \forall i \in I, j \in J, \\
y_{rj}^l(\alpha) &= \bar{y}_{rj} - t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\psi}_{rj}}{\sqrt{p}}, & y_{rj}^u(\alpha) &= \bar{y}_{rj} - t_{\left(\frac{\alpha}{2}\right)} \frac{\bar{\psi}_{rj}}{\sqrt{p}}, & \forall r \in O, j \in J.
\end{aligned}$$

Where

- $\bar{\alpha}_{ij}$ : input mean
- $\bar{\sigma}_{ij}^2$ : input variance
- $\bar{y}_{rj}$ : output mean
- $\bar{\psi}_{rj}^2$ : output variance
- $t_{\left(\frac{\alpha}{2}\right)}$ : T-distribution with  $p - 1$  degree of freedom

Then we get the following model to estimate the upper and lower bound of the efficiency of  $DMU_{n+1}$  with  $(1 - \alpha)\%$  confidence interval using optimistic and pessimistic viewpoints; [13](#)

$$\begin{aligned}
\varphi_{n+1}^u &= \max \varphi \\
\text{s.t.} \quad & \sum_{j \in J} \mu_j \alpha_{ij}^l(\alpha) + \mu_{n+1} \alpha_{i(n+1)}^u(\alpha) \leq \alpha_{in+1}^u(\alpha), \quad \forall i \in I, \\
& \sum_{j \in J} \mu_j y_{rj}^u(\alpha) + \mu_{n+1} y_{rn+1}^l(\alpha) \geq \phi y_{r(n+1)}^l(\alpha), \quad \forall r \in O \\
& \delta_1 \left( \sum_{j=1}^n \mu_j + \mu_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& \nu \geq 0, \mu_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned} \tag{5.6}$$

$$\begin{aligned}
\varphi_{n+1}^l &= \max \varphi \\
\text{s.t.} \quad & \sum_{j \in J} \mu_j \alpha_{ij}^u(\alpha) + \mu_{n+1} \alpha_{i(n+1)}^l(\alpha) \leq \alpha_{in+1}^l(\alpha), \quad \forall i \in I, \\
& \sum_{j \in J} \mu_j y_{rj}^l(\alpha) + \mu_{n+1} y_{r(n+1)}^u(\alpha) \geq \phi y_{r(n+1)}^u(\alpha), \quad \forall r \in O \\
& \delta_1 \left( \sum_{j=1}^n \mu_j + \mu_{n+1} + \delta_2 (-1)^{\delta_3} \nu \right) = \delta_1, \\
& \nu \geq 0, \mu_j \geq 0, j = 1, \dots, n. \quad \delta_i \in \{0, 1\}; i = 1, 2, 3
\end{aligned}$$

The models [\(5.6\)](#) estimate the upper and lower bounds of efficiency for DMUn+1 within a  $(1 - \alpha)\%$  confidence interval. The efficiency index for  $DMU_{n+1}$ ,  $\phi_{n+1}(1 - \alpha)$  falls between these bounds. Now we do the following check to find output levels.

- Optimal value of model [\(5.4\)](#) checked.
- Output levels set equal to original levels
- Efficiency bounds for DMUn+1 considered.
- If the value doesn't lie within bounds, then we'll apply the method mentioned above.

- Then will find output levels while keeping the efficiency index( $\phi_o(\alpha)$ ) of the model (4.4) constant
- Retain initial efficiency value at a specific level of significance

Now final remark is when we have a symmetric error in both input and output, then we will solve both problems in a similar manner as we have converted the Chance-constrained model in Chapter 5 by incorporating a symmetric error structure to convert it into an LP problem. Below Table show LP and MOLP form of above described Chance-constrained models (5.1) and (5.4) and SMOP (5.2) and (5.5),

	Input Estimation	Output Estimation
<b>LP Form</b>	$\theta_o^{n+1} = \min \theta$ <p>s.t.</p> $\sum_{j=1}^n \lambda_j x_{ij} + \lambda_{n+1} \alpha_{io} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq \theta \alpha_{io} \quad i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j y_{rj} + \lambda_{n+1} \beta_{ro} + \Phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq \beta_{ro} \quad r = 1, \dots, s$ $\sum_{j=1}^n \lambda_j a_{ij} + \lambda_{n+1} \bar{a}_{io} - \theta \bar{a}_{io} = p_i^+ - p_i^- \quad i = 1, 2, \dots, m$ $\sum_{j=1}^n \lambda_j b_{rj} + \lambda_{n+1} \bar{b}_{ro} - \bar{b}_{ro} = q_r^+ - q_r^- \quad r = 1, 2, \dots, s$ $p_i^+ \geq 0 \quad p_i^- \geq 0 \quad q_r^+ \geq 0 \quad q_r^- \geq 0 \quad i = 1, 2, \dots, m \quad r = 1, 2, \dots, s$ $\lambda_j \geq 0 \quad ; j = 1, 2, \dots, n$	$\phi_o^{n+1} = \min \phi$ <p>s.t.</p> $\sum_{j=1}^n \lambda_j x_{ij} + \lambda_{n+1} \alpha_{io} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq \alpha_{io} \quad i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j y_{rj} + \lambda_{n+1} \beta_{ro} + \Phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq \phi \beta_{ro} \quad r = 1, \dots, s$ $\sum_{j=1}^n \lambda_j a_{ij} + \lambda_{n+1} \bar{a}_{io} - \bar{a}_{io} = p_i^+ - p_i^- \quad i = 1, 2, \dots, m$ $\sum_{j=1}^n \lambda_j b_{rj} + \lambda_{n+1} \bar{b}_{ro} - \phi \bar{b}_{ro} = q_r^+ - q_r^- \quad r = 1, 2, \dots, s$ $p_i^+ \geq 0 \quad p_i^- \geq 0 \quad q_r^+ \geq 0 \quad q_r^- \geq 0 \quad i = 1, 2, \dots, m \quad r = 1, 2, \dots, s$ $\lambda_j \geq 0 \quad ; j = 1, 2, \dots, n$
<b>MOLP</b>	$\min(\tilde{\alpha}_{1o}, \tilde{\alpha}_{2o}, \dots, \tilde{\alpha}_{mo})$ <p>s.t.</p> $\sum_{j=1}^n \lambda_j x_{ij} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq \theta \alpha_{io} \quad i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j y_{rj} + \Phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq \beta_{ro} \quad r = 1, \dots, s$ $\alpha_{io} - \Phi^{-1}(\alpha) \bar{\sigma}(t_i^+ + t_i^-) \geq x_{io} \quad i = 1, 2, \dots, m$ $\sum_{j=1}^n \lambda_j a_{ij} - \theta \bar{a}_{io} = p_i^+ - p_i^- \quad i = 1, 2, \dots, m$ $\sum_{j=1}^n \lambda_j b_{rj} - \bar{b}_{ro} = q_r^+ - q_r^- \quad r = 1, 2, \dots, s$ $\bar{a}_{io} - \alpha_{io} = t_i^+ + t_i^- \quad i = 1, 2, \dots, m$ $t_i^+ \geq 0 \quad t_i^- \geq 0 \quad p_i^+ \geq 0 \quad p_i^- \geq 0 \quad i = 1, 2, \dots, m$ $q_r^+ \geq 0 \quad q_r^- \geq 0 \quad r = 1, 2, \dots, s \quad \lambda_j \geq 0 \quad ; j = 1, 2, \dots, n$	$\max(\tilde{\beta}_{1o}, \tilde{\beta}_{2o}, \dots, \tilde{\beta}_{so})$ <p>s.t.</p> $\sum_{j=1}^n \lambda_j x_{ij} - \Phi^{-1}(\alpha) \bar{\sigma}(p_i^+ + p_i^-) \leq \alpha_{io} \quad i = 1, \dots, m$ $\sum_{j=1}^n \lambda_j y_{rj} + \Phi^{-1}(\alpha) \bar{\sigma}(q_r^+ + q_r^-) \geq \phi \beta_{ro} \quad r = 1, \dots, s$ $\beta_{ro} - \Phi^{-1}(\alpha) \bar{\sigma}(t_r^+ + t_r^-) \geq y_{ro} \quad r = 1, 2, \dots, s$ $\sum_{j=1}^n \lambda_j a_{ij} - \bar{a}_{io} = p_i^+ - p_i^- \quad i = 1, 2, \dots, m$ $\sum_{j=1}^n \lambda_j b_{rj} - \phi \bar{b}_{ro} = q_r^+ - q_r^- \quad r = 1, 2, \dots, s$ $\bar{b}_{ro} - b_{ro} = t_r^+ + t_r^- \quad r = 1, 2, \dots, s$ $\lambda_j \geq 0 \quad ; j = 1, 2, \dots, n \quad p_i^+ \geq 0 \quad p_i^- \geq 0 \quad i = 1, 2, \dots, m$ $q_r^+ \geq 0 \quad q_r^- \geq 0 \quad t_r^+ \geq 0 \quad t_r^- \geq 0 \quad r = 1, 2, \dots, s$

Table 5.1: LP and MLOP form of Chance-constrained Inverse DEA in significance level  $\alpha$

## 5.2 Application in Banking Sector

This section demonstrates the application of the described methodology in the context of the banking industry by utilizing a dataset from 20 branches of an Iranian commercial bank. The goal is to validate the practicality of the research objectives. The dataset, which comprises three input factors and five output factors, has been obtained from the work of Behzadi and Mirbolouki and is displayed in Table (B.3) and (B.4). the suggested methodology can be applied to various domains, not just the banking sector.

In the existing literature, there are two dominant approaches for selecting input and output factors: the production approach and the intermediation approach. For this study, we have chosen to employ the intermediation approach.

- **Inputs:**

- Personal rate ( $\tilde{x}_1$ ): A weighted combination of personal qualifications, quantity, education, and other bank branch factors, considered as the cost input.
- Payable benefits ( $\tilde{x}_2$ ): The benefits payable to customers for all deposits in each branch.
- Delayed requisitions ( $\tilde{x}_3$ ): Delays in repaying loans and other facilities in each branch.

- **Outputs:**

- Facilities ( $\tilde{y}_1$ ): The sum of business and individual loans in each branch.
- Amount of deposits ( $\tilde{y}_2$ ): The value of various deposits, including current, short-term, and long-term accounts in each branch.
- Received benefits ( $\tilde{y}_3$ ): The benefits received from the total loans and facilities.
- Received commission ( $\tilde{y}_4$ ): The total commission received from all banking actions, issuance guarantees, money transfers, and others in each branch.
- Other resources of deposits ( $\tilde{y}_5$ ): Additional sources of deposits in each branch.

The cost input is determined by a weighted combination of factors such as personal qualifications, quantity, education level, and the influence of other bank branches. The benefits paid out to customers are derived from all deposits made in each branch. Delayed claims represent instances of late repayment of loans and other facilities within each branch. The output factor "facilities" consists of the aggregated amount of business loans and individual loans within each branch. The total amount of deposits is represented by the combined value of various deposit types, such as current accounts, short-term accounts, and long-term accounts in each branch. The benefits received are determined by the overall loans and facilities, while the commissions acquired are the aggregate amount of all banking activities, including the issuance of guarantees, money transfers, and other related operations within each branch.

The table(4.6) shows the calculation of each DMUs under CRS technology at various significance levels. From the table(4.6) it is clear that DMU10 is efficient at the significance level  $\alpha = 0.05$  as  $\theta_{10}(0.05) = 1$ ; For DMU10, we aim to investigate the possible increases in input levels while keeping its efficiency index unchanged at the significance level of 0.05. The input levels include personal rate, payable benefits, and delayed requisitions.

The below table shows the change in output levels of DMU10 ;

Table 5.2: The percentage of expected increases in the output levels for DMU10.

Outputs	Facilities		Amount of Deposits		Received Benefits		Received Commission		Other Resources of Deposits	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Old Output ( $\tilde{Y}_{10}$ )	1229.1	69.02	194.58	17.35	25.91	12.3	86.76	22.1	600.6	86.34
New Output ( $\beta_{10}$ )	1309.1	73.02	204.58	18.35	27.11	12.8	91.76	22.3	615.6	87.34
Expected Increase ( $\Delta\tilde{Y}_{10}$ )	80.0	4.0	10.0	1.0	1.2	0.5	5.0	0.2	15.0	1.0
Percentage Changes	6.51%	5.80%	5.14%	5.76%	4.63%	4.07%	5.76%	0.90%	2.50%	1.16%

Now we will employ MOLP in input orientation as in table(5.1) to analyze the change in input data of DMU10 in the level of significance 0.05 and result is shown in Table(5.3) and python code for calculation is attached in Appendix section.

Table(5.3) offers two scenarios for decision-makers to select the most suitable strategy for expanding DMU10. These scenarios ensure that the efficiency score remains equal to one at the 0.05 significance level, based on the input data of DMU10. By increasing the output levels of DMU10 from Old Output ( $\tilde{Y}_{10}$ ) to New Output ( $\tilde{\beta}_{10}$ ), the decision-makers should do the following operations to maintain the efficiency score i.e.  $\theta_{10} = 1$ :

Table 5.3: The percentage of necessary increases in the inputs for DMU10.

Inputs	Personal rate ( $\tilde{\alpha}_{10}^1$ )		Payable benefits ( $\tilde{\alpha}_{10}^2$ )		Delayed requisitions ( $\tilde{\alpha}_{10}^3$ )	
	Mean	Variance	Mean	Variance	Mean	Variance
Initial input DMU10	19.88	0.25	46.41	53.1	12.81	0.38
The first SP solution ( $\tilde{\alpha}_1$ )	20.98	0.28	49.04	57.33	14.88	0.77
The expected increase ( $\Delta\tilde{X}_{10}$ )	1.10	0.03	2.63	4.23	2.07	0.39
Percentage increases	5.53%	12.00%	5.67%	7.97%	16.17%	102.63%
The second SP solution ( $\tilde{\alpha}_2$ )	20.87	0.29	48.62	57.06	15.73	1.39
The expected increase ( $\Delta\tilde{X}_{10}$ )	0.99	0.04	2.21	3.96	2.92	1.01
Percentage increases	4.98%	16.00%	4.76%	7.46%	22.79%	266.32%

1. For the personal rate input level ( $\alpha_{10}^1$ ):
  - In Scenario 1, the mean should increase by 5.53% and the variance by 12.00%.
  - In Scenario 2, the mean should increase by 4.98% and the variance by 16.00%.
2. For the payable benefits input level ( $\alpha_{10}^2$ ):
  - In Scenario 1, the mean should increase by 5.67% and the variance by 7.97%.
  - In Scenario 2, the mean should increase by 4.76% and the variance by 7.46%.
3. For the delayed requisitions input level ( $\alpha_{10}^3$ ):
  - In Scenario 1, the mean should increase by 16.17% and the variance by 102.63%.
  - In Scenario 2, the mean should increase by 22.79% and the variance by 266.32%.

Decision-makers can review the two scenarios and choose the most appropriate one for expanding DMU10, considering the specific context and constraints of their organization. Each scenario suggests different degrees of input level increases, allowing decision-makers to weigh the trade-offs and select the most suitable option for achieving growth while maintaining efficiency.



Now we will analyze a similar case in the case of an inefficient DMU18. Below Table illustrates the data for DMU18, which is currently inefficient with an efficiency index of 0.53. The table displays the changes in output levels for DMU18, assuming that its efficiency index remains unchanged at the 0.53 level while the output levels increase from Old Output ( $\tilde{Y}_{18}$ ) to New Output ( $\tilde{\beta}_{18}$ ). The expected percentage increases in the various outputs are also provided in the table.

Outputs	Facilities		Amount of deposits		Received benefits		Received commission		Other resources of deposits	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Old output ( $\tilde{Y}_{18}$ )	259.21	34.21	81.779	21.65	5.212	1.94	8.021	3.52	107.9	13.07
New output ( $\tilde{\beta}_{18}$ )	271.52	36.19	85.219	22.38	5.432	2.05	8.351	3.68	113.4	13.35
The expected increase ( $\Delta\tilde{Y}_{18}$ )	12.31	1.98	3.44	0.73	0.22	0.11	0.33	0.16	5.5	0.28
Percentage changes	4.75%	5.78%	4.21%	3.37%	4.22%	5.67%	4.12%	4.55%	5.10%	2.14%

Table 5.4: The percentage of expected increase in the outputs for DMU18.

Table (5.5) demonstrates the necessary increases in input levels for DMU18 in order to maintain its current efficiency index while increasing the output levels. Two scenarios are presented, the first SP solution ( $\tilde{\alpha}_{18}$ ) and the second SP solution ( $\tilde{\alpha}_{218}$ ), each offering different percentages of necessary input level increases.

Table 5.5: The percentage of necessary increases in the inputs for DMU18.

Inputs	Personal rate		Payable benefits		Delayed requisitions	
	Mean	Variance	Mean	Variance	Mean	Variance
Old input ( $\tilde{X}_{18}$ )	13.71	0.18	40.32	51.4	10.88	0.12
First SP solution ( $\tilde{\alpha}_{18}$ )	15.27046	1.039409	40.0856	51.46431	26.1024	0.66521
Expected increase ( $\Delta\tilde{X}_{18}^1$ )	1.56046	0.859409	-0.2344	0.06431	15.2224	0.54521
Percentage increases	11.38%	477.45%	-0.58%	0.13%	139.94%	454.34%
Second SP solution ( $\tilde{\alpha}_{218}$ )	21.1160	0.201329	36.8938	25.14718	34.8968	11.06882
Expected increase ( $\Delta\tilde{X}_{18}^2$ )	7.4060	0.021329	-3.4262	-26.25282	24.0168	10.94882
Percentage increases	54.01%	11.85%	-8.50%	-51.12%	220.70%	9124.02%

These two scenarios offer decision-makers alternative strategies for expanding DMU18 while preserving its efficiency index at the 0.53 level. The required input level adjustments for each scenario are as follows:

1. For the personal rate input level :

- In Scenario 1, the mean should increase by 11.38% and the variance by 477.45%.
  - In Scenario 2, the mean should increase by 54.01% and the variance by 11.85%.
2. For the payable benefits input level :
- In Scenario 1, the mean should decrease by 0.58% and the variance increase by 0.13%.
  - In Scenario 2, the mean should decrease by 8.50% and the variance decrease by 51.12%.
3. For the delayed requisitions input level :
- In Scenario 1, the mean should increase by 139.94% and the variance by 454.34%.
  - In Scenario 2, the mean should increase by 220.70% and the variance by 9124.02%.

Decision-makers can evaluate these scenarios and select the most suitable one for DMU18 expansion, considering their organization's specific context and constraints. Each scenario suggests varying input level increases, enabling decision-makers to assess the trade-offs and choose the best option for growth while preserving efficiency.

In summary, Tables (5.4) and (5.5) provides decision-makers with valuable information on the changes in output levels for DMU18 and the necessary increases in input levels under two different scenarios. This information allows decision-makers to choose the most appropriate strategy for expanding DMU18 while maintaining its efficiency index at the 0.53 level.

## Conclusion

In this Chapter, we expanded the InvDEA approach for input/output estimation with stochastic data, presenting a new optimality concept in multi-objective optimization problems [13]. The application of this methodology in the banking sector

demonstrated its effectiveness in achieving desired efficiency levels. This work paves the way for future research in dynamic and network DEA frameworks and practical applications in various industries [13].

# Chapter 6

## Conclusion and Future Research

### 6.1 Summary of Contributions:

This thesis has presented a comprehensive exploration of classical DEA models, inverse DEA, and stochastic DEA, culminating in the development of new inverse DEA models that incorporate stochastic data. The main contributions of this research are:

- A detailed introduction to classical DEA models, including the CCR and BCC models, along with their Python Codes for Practical implementations.
- An examination of inverse DEA and its application to efficiency analysis in a case study of 15 retail stores with hypothetical data.
- A thorough introduction to stochastic DEA models and their real-world applications, considering the uncertainty inherent in data.
- The development and application of new inverse DEA models that account for stochastic data, which enhances the applicability of DEA models in uncertain environments.
- Python code implementation for each DEA model, provided in the Appendix, to aid in the practical application and replication of the models.

### 6.2 Future Research

This thesis lays the foundation for several potential research directions, which can further enhance and refine the methodologies and applications of DEA models in

uncertain environments. Future research opportunities include:

- Investigating various types of uncertainty in DEA models, such as interval or fuzzy data, to broaden their applicability in diverse real-world situations.
- Developing more efficient algorithms to solve large-scale problems and high-dimensional data in DEA models, improving their scalability and computational efficiency.
- Incorporating machine learning and artificial intelligence techniques into DEA models, aiming to enhance their performance, predictive capabilities, and model selection processes.
- Examining the robustness and sensitivity of the proposed DEA models against outliers and data errors, and devising methods to mitigate their potential impact on model performance.
- Conducting comparative studies of the proposed DEA models with other efficiency measurement techniques, both deterministic and stochastic, to further validate their usefulness and applicability in real-world settings.
- Applying the developed DEA models to a variety of industries and sectors, such as healthcare, education, and energy, to demonstrate their practical implications and effectiveness in different contexts.

By pursuing these research avenues, future studies can continue to enhance and refine DEA models, particularly in the context of uncertain environments, ultimately leading to more robust and accurate efficiency analyses across various industries and settings.

# Bibliography

- [1] Alireza Amirteimoori et al. *Stochastic Benchmarking*. Springer, 2022.
- [2] Rajiv D Banker, Abraham Charnes, and William Wager Cooper. “Some models for estimating technical and scale inefficiencies in data envelopment analysis”. In: *Management science* 30.9 (1984), pp. 1078–1092.
- [3] MH Behzadi and M Mirbolouki. “Symmetric error structure in stochastic DEA”. In: (2012).
- [4] Abraham Charnes, William W Cooper, and Edwardo Rhodes. “Measuring the efficiency of decision making units”. In: *European journal of operational research* 2.6 (1978), pp. 429–444.
- [5] Abraham Charnes et al. “Data envelopment analysis theory, methodology and applications”. In: *Journal of the Operational Research society* 48.3 (1997), pp. 332–333.
- [6] Abraham Charnes et al. “Foundations of data envelopment analysis for Pareto-Koopmans efficient empirical production functions”. In: *Journal of econometrics* 30.1-2 (1985), pp. 91–107.
- [7] Wade D Cook and Larry M Seiford. “Data envelopment analysis (DEA)—Thirty years on”. In: *European journal of operational research* 192.1 (2009), pp. 1–17.
- [8] William W Cooper, Lawrence M Seiford, and Kaoru Tone. *Data envelopment analysis: a comprehensive text with models, applications, references, and DEA-solver software*. Vol. 2. Springer, 2007.
- [9] William W Cooper et al. “Chance constrained programming approaches to technical efficiencies and inefficiencies in stochastic data envelopment analysis”. In: *Journal of the Operational Research Society* 53.12 (2002), pp. 1347–1356.
- [10] Andreas Dellnitz, Andreas Kleine, and Wilhelm Rödder. “CCR or BCC: what if we are in the wrong model?” In: *Journal of Business Economics* 88 (2018), pp. 831–850.
- [11] Basma E El-Demerdash, Ihab A El-Khodary, and Assem A Tharwat. “Developing a stochastic input-oriented data envelopment analysis (SIODEA) model”. In: *International Journal of Advanced Computer Science and Applications* 4.4 (2013).

- [12] Said Gattoufi, Gholam R Amin, and Ali Emrouznejad. “A new inverse DEA method for merging banks”. In: *IMA Journal of Management Mathematics* 25.1 (2014), pp. 73–87.
- [13] Ali Ghomi et al. “Inverse data envelopment analysis with stochastic data”. In: *RAIRO-Operations Research* 55.5 (2021), pp. 2739–2762.
- [14] Kenneth C Land, CA Knox Lovell, and Sten Thore. “Chance-constrained data envelopment analysis”. In: *Managerial and decision economics* 14.6 (1993), pp. 541–554.
- [15] Ankita Panwar et al. “A Review on the 40 Years of Existence of Data Envelopment Analysis Models: Historic Development and Current Trends”. In: *Archives of Computational Methods in Engineering* (2022), pp. 1–30.
- [16] Tahere Sayar, Mojtaba Ghiyasi, and Jafar Fathali. “New inverse DEA models for budgeting and planning”. In: *RAIRO-Operations Research* 55.3 (2021), pp. 1933–1948.
- [17] Toshiyuki Sueyoshi and Mika Goto. “DEA radial and non-radial models for unified efficiency under natural and managerial disposability: theoretical extension by strong complementary slackness conditions”. In: *Energy Economics* 34.3 (2012), pp. 700–713.
- [18] Kaoru Tone. “A slacks-based measure of efficiency in data envelopment analysis”. In: *European journal of operational research* 130.3 (2001), pp. 498–509.
- [19] Quanling Wei, Jianzhong Zhang, and Xiangsun Zhang. “An inverse DEA model for inputs/outputs estimate”. In: *European Journal of Operational Research* 121.1 (2000), pp. 151–163.
- [20] M. Wen. *Uncertain Data Envelopment Analysis*. Uncertainty and Operations Research. Springer Berlin Heidelberg, 2016. ISBN: 9783662525814. URL: [https://books.google.co.in/books?id=\\\_99QvgAACAAJ](https://books.google.co.in/books?id=\_99QvgAACAAJ)
- [21] GuoJun Zhang and JinChuan Cui. “A general inverse DEA model for non-radial DEA”. In: *Computers & Industrial Engineering* 142 (2020), p. 106368.

# Appendix A

## Code listings

### A.1 Python Codes for CCR, BCC, ADDITIVE, and SBM models

All codes use Gurobi Python and data will be in the format of a NumPy array. And by importing necessary libraries like;

```
import numpy as np
import pandas as pd
from gurobipy import Model, GRB, quicksum and set data as numpy array
```

#### A.1.1 CCR Codes

Codes for the First Phase for calculating efficiencies of the CCR model:

Listing A.1: CCR Phase 1 code

```
1 input_data = np.array([])
2 output_data = np.array([])
3 def input_efficiencyCCR(input_data, output_data):
4     n, m = input_data.shape
5     n, s = output_data.shape
6     efficiencies = []
7
8     for o in range(n):
9         model = Model("InputEfficiency")
10        model.setParam(GRB.Param.OutputFlag, 0)
11
12        lambdas = [model.addVar(lb=0, name=f"lambda_{j}") for
```



```

13         j in range(n)]
14     theta = model.addVar(lb=0, name="theta")
15
16     model.setObjective(theta, GRB.MINIMIZE)
17
18     for i in range(m):
19         model.addConstr(quicksum(lambdas[j] *
20             input_data[j, i] for j in range(n)) <= theta *
21             input_data[o, i])
22
23     for r in range(s):
24         model.addConstr(quicksum(lambdas[j] *
25             output_data[j, r] for j in range(n)) >=
26             output_data[o, r])
27
28     model.optimize()
29
30     efficiencies.append(model.objVal)
31
32     return efficiencies
33
34 input_efficiencies = input_efficiencyCCR(input_data,
35     output_data)
36
37 print(input_efficiencies)

```

Now below codes is for maximizing slacks of the CCR model, which is Phase 2 of the CCR model and we use above obtained input efficiencies in Phase2:

Listing A.2: CCR Phase 2 code

```

1 def phase_twoCCR(input_data, output_data, input_efficiencies):
2     n, m = input_data.shape
3     n, s = output_data.shape
4     results = []
5
6     for o in range(n):
7         theta_star = input_efficiencies[o]
8         model = Model("Phase2")
9         model.setParam(GRB.Param.OutputFlag, 0)
10
11         lambdas = [model.addVar(lb=0, name=f"lambda_{j}") for
12             j in range(n)]

```

```

12     S_minus = [model.addVar(lb=0, name=f"S_minus_{i}")
13                 for i in range(m)]
14     S_plus = [model.addVar(lb=0, name=f"S_plus_{r}") for
15                 r in range(s)]
16
17     model.setObjective(quicksum(S_minus) +
18                       quicksum(S_plus), GRB.MAXIMIZE)
19
20     for i in range(m):
21         model.addConstr(quicksum(lambdas[j] *
22                                 input_data[j, i] for j in range(n)) +
23                         S_minus[i] == theta_star * input_data[o, i])
24
25     for r in range(s):
26         model.addConstr(quicksum(lambdas[j] *
27                                 output_data[j, r] for j in range(n)) -
28                         S_plus[r] >= output_data[o, r])
29
30     model.optimize()
31
32     results.append({
33         'DMU': o,
34         'efficiency': theta_star,
35         'slacks_minus': [S_minus[i].x for i in range(m)],
36         'slacks_plus': [S_plus[r].x for r in range(s)],
37         'Lambda': [lambdas[j].x for j in range(n)]
38     })
39
40     return results
41
42 phase_two_results = phase_twoCCR(input_data, output_data,
43                                 input_efficiencies)
44 results_df = pd.DataFrame(phase_two_results)
45 print(results_df)

```

## A.1.2 BCC Codes

Below code will solve Phase 1 and Phase 2 of BCC Models:

```

1     #Phase one of BCC model
2     def input_efficiencyBCC(input_data, output_data):
3         n, m = input_data.shape
4         n, s = output_data.shape
5         efficiencies = []
6
7         for o in range(n):
8             model = Model("InputEfficiency")
9             model.setParam(GRB.Param.OutputFlag, 0)
10
11            lambdas = [model.addVar(lb=0, name=f"lambda_{j}") for
12                        j in range(n)]
13            theta = model.addVar(lb=0, name="theta")
14
15            model.setObjective(theta, GRB.MINIMIZE)
16
17            for i in range(m):
18                model.addConstr(quicksum(lambdas[j] *
19                                        input_data[j, i] for j in range(n)) <= theta *
20                                input_data[o, i])
21
22            for r in range(s):
23                model.addConstr(quicksum(lambdas[j] *
24                                        output_data[j, r] for j in range(n)) >=
25                                output_data[o, r])
26
27            model.addConstr(quicksum(lambdas[j] for j in
28                                    range(n)) ==1)
29
30            model.optimize()
31
32            efficiencies.append(model.objVal)
33
34        return efficiencies
35
36    input_efficiencies = input_efficiencyBCC(input_data,
37        output_data)
38    print(input_efficiencies)

```

```

34
35 #phase two of BCC model
36 def phase_twoBCC(input_data, output_data, input_efficiencies):
37     n, m = input_data.shape
38     n, s = output_data.shape
39     results = []
40
41     for o in range(n):
42         theta_star = input_efficiencies[o]
43         model = Model("Phase2")
44         model.setParam(GRB.Param.OutputFlag, 0)
45
46         lambdas = [model.addVar(lb=0, name=f"lambda_{j}") for
47                     j in range(n)]
48         S_minus = [model.addVar(lb=0, name=f"S_minus_{i}")
49                     for i in range(m)]
50         S_plus = [model.addVar(lb=0, name=f"S_plus_{r}") for
51                       r in range(s)]
52
53         model.setObjective(quicksum(S_minus) +
54                             quicksum(S_plus), GRB.MAXIMIZE)
55
56         for i in range(m):
57             model.addConstr(quicksum(lambdas[j] *
58                                     input_data[j, i] for j in range(n)) +
59                             S_minus[i] == theta_star * input_data[o, i])
60
61         for r in range(s):
62             model.addConstr(quicksum(lambdas[j] *
63                                     output_data[j, r] for j in range(n)) -
64                             S_plus[r] >= output_data[o, r])
65
66         model.addConstr(quicksum(lambdas[j] for j in
67                                 range(n)) ==1)
68         model.optimize()
69
70         results.append({
71             'DMU': o,
72             'efficiency': theta_star,
73             'slacks_minus': [S_minus[i].x for i in range(m)],
74             'slacks_plus': [S_plus[r].x for r in range(s)],

```

```

66         'Lambda': [lambdas[j].x for j in range(n)]
67     })
68
69     return results
70
71 phase_two_results = phase_twoBCC(input_data, output_data,
72     input_efficiencies)
73 results_df = pd.DataFrame(phase_two_results)
74 print(results_df)

```

### A.1.3 Additive model Code

```

1 def additive_model(input_data, output_data):
2     n, m = input_data.shape
3     n, s = output_data.shape
4     results = []
5
6     for o in range(n):
7         model = Model("AdditiveModel")
8         model.setParam(GRB.Param.OutputFlag, 0)
9
10        lambdas = [model.addVar(lb=0, name=f"lambda_{j}") for
11            j in range(n)]
12        S_minus = [model.addVar(lb=0, name=f"S_minus_{i}")
13            for i in range(m)]
14        S_plus = [model.addVar(lb=0, name=f"S_plus_{r}") for
15            r in range(s)]
16
17        model.setObjective(quicksum(S_minus) +
18            quicksum(S_plus), GRB.MAXIMIZE)
19
20        for i in range(m):
21            model.addConstr(quicksum(lambdas[j] *
22                input_data[j, i] for j in range(n)) +
23                S_minus[i] == input_data[o, i])
24
25        for r in range(s):
26            model.addConstr(quicksum(lambdas[j] *
27                output_data[j, r] for j in range(n)) -

```

```

21         S_plus[r] == output_data[o, r])
22     model.addConstr(quicksum(lambdas[j] for j in
23                             range(n)) == 1)
24
25     model.optimize()
26
27     results.append({
28         'DMU': o,
29         'slacks_minus': [S_minus[i].x for i in range(m)],
30         'slacks_plus': [S_plus[r].x for r in range(s)],
31         'Lambda': [lambdas[j].x for j in range(n)]
32     })
33
34     return results
35
36 additive_model_results = additive_model(input_data,
37                                         output_data)
38 results_df = pd.DataFrame(additive_model_results)
39 print(results_df)

```

#### A.1.4 SBM codes

```

1 def sbm_model(input_data, output_data):
2     n, m = input_data.shape
3     n, s = output_data.shape
4     results = []
5
6     for o in range(n):
7         model = Model("OriginalSBM_Model")
8         model.setParam(GRB.Param.OutputFlag, 0)
9
10        Lambda = [model.addVar(lb=0, name=f"Lambda_{j}") for
11                  j in range(n)]
12        S_minus = [model.addVar(lb=0, name=f"S_minus_{i}")
13                  for i in range(m)]
14        S_plus = [model.addVar(lb=0, name=f"S_plus_{r}") for
15                  r in range(s)]

```

```

13     t = model.addVar(lb=0, name="t")
14
15     model.setObjective(t - quicksum(S_minus[i] /
16         input_data[o, i] for i in range(m)) / m,
17         GRB.MINIMIZE)
18
19     model.addConstr(t + quicksum(S_plus[r] /
20         output_data[o, r] for r in range(s)) / s == 1)
21
22     for i in range(m):
23         model.addConstr(quicksum(Lambda[j] *
24             input_data[j, i] for j in range(n)) == t *
25             input_data[o, i] - S_minus[i])
26
27     for r in range(s):
28         model.addConstr(quicksum(Lambda[j] *
29             output_data[j, r] for j in range(n)) == t *
30             output_data[o, r] + S_plus[r])
31
32     model.optimize()
33
34     rho_star = model.objVal
35     t_star = t.x
36     lambda_star = [Lambda[j].x / t_star for j in range(n)]
37     s_minus_star = [S_minus[i].x / t_star for i in
38         range(m)]
39     s_plus_star = [S_plus[r].x / t_star for r in range(s)]
40
41     results.append({
42         'DMU': o,
43         'rho': rho_star,
44         'lambda': lambda_star,
45         's_minus': s_minus_star,
46         's_plus': s_plus_star
47     })
48
49     return results
50
51 sbm_model_results = sbm_model(input_data, output_data)
52 results_df = pd.DataFrame(sbm_model_results)
53 print(results_df)

```

---

## A.2 Python Codes for inverse DEA for data of 15 Stores

This Code calculates the Input-Output efficiencies of stores

Listing A.3: Efficiency Calculation code for Stores

```
1 import numpy as np
2 from gurobipy import Model, quicksum, GRB
3 input_data = np.array([ [500, 1200], [600, 1500], [450,
4     1000], [800, 2000], [900, 2200],
5     [400, 1100], [550, 1300], [650, 1700], [300, 900],
6     [750, 1800],
7     [700, 1600], [850, 2100], [350, 950], [950, 2300],
8     [1000, 2400], ])
9 output_data = np.array([ [500000, 0.80],[600000,
10     0.82],[400000, 0.75],[900000, 0.85], [1000000, 0.90],
11     [350000, 0.70], [450000, 0.78], [700000, 0.84], [300000,
12     0.68], [800000, 0.83], [750000, 0.81], [950000, 0.88],
13     [320000, 0.72], [1100000, 0.92], [1200000,0.95],])
14 import pandas as pd
15 from gurobipy import Model, GRB, quicksum
16
17 def efficiency(input_data, output_data):
18     n, m = input_data.shape
19     n, s = output_data.shape
20     input_efficiencies = []
21     output_efficiencies = []
22     for o in range(n):
23         input_model = Model("input_efficiency")
24         input_model.setParam(GRB.Param.OutputFlag, 0)
25         lambdas_input = [input_model.addVar(lb=0,
26             name=f"lambda_input_{j}") for j in range(m)]
27         theta_input = input_model.addVar(lb=0,
28             name="theta_input")
29         input_model.setObjective(theta_input, GRB.MINIMIZE)
```



```

22     input_model.addConstr(quicksum(lambdas_input[j] for j
23         in range(n)) == 1)
24     for i in range(m):
25         input_model.addConstr(quicksum(lambdas_input[j] *
26             input_data[j, i] for j in range(n)) <=
27             theta_input * input_data[o, i])
28     for r in range(s):
29         input_model.addConstr(quicksum(lambdas_input[j] *
30             output_data[j, r] for j in range(n)) >=
31             output_data[o, r])
32     input_model.optimize()
33     input_efficiency = theta_input.X
34     input_efficiencies.append(input_efficiency)
35     output_model = Model("output_efficiency")
36     output_model.setParam(GRB.Param.OutputFlag, 0)
37     lambdas_output = [output_model.addVar(lb=0,
38         name=f"lambda_output_{j}") for j in range(n)]
39     phi = output_model.addVar(lb=0, name="phi")
40     output_model.setObjective(phi, GRB.MAXIMIZE)
41     output_model.addConstr(quicksum(lambdas_output[j] for
42         j in range(n)) == 1)
43     for i in range(m):
44         output_model.addConstr(quicksum(lambdas_output[j]
45             * input_data[j, i] for j in range(n)) <=
46             input_data[o, i])
47     for r in range(s):
48         output_model.addConstr(quicksum(lambdas_output[j]
49             * output_data[j, r] for j in range(n)) >= phi
50             * output_data[o, r])
51     output_model.optimize()
52     output_efficiency = phi.X
53     output_efficiencies.append(output_efficiency)
54     return input_efficiencies, output_efficiencies
55 Stores_names = [ 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I',
56     'J', 'K', 'L', 'M', 'N', 'O']
57 input_efficiencies, output_efficiencies =
58     efficiency(input_data, output_data)
59 result = pd.DataFrame({'Supermarket': supermarket_names,
60     'Input_Efficiencies': input_efficiencies,
61     'Output_Efficiencies': output_efficiencies})
62 print(result)

```

Below codes solve the income and budget based inverse DEA;

Listing A.4: Income and Budget Based python codes for Inverse DEA

```
1 def income_based_dea(input_data, output_data,
2   input_efficiency, input_prices, output_prices, A):
3     n = len(input_data)
4     m = len(input_data[0])
5     s = len(output_data[0])
6     results = []
7     for k in range(n):
8         model = Model(f"Income_based_DEA_Supermarket_{k + 1}")
9         lambdas = model.addVars(n, name="lambda", lb=0)
10        dx = model.addVars(m, name="dx", lb=0)
11        dy = model.addVars(s, name="dy", lb=0)
12        model.setObjective(quicksum(input_prices[i] * dx[i]
13            for i in range(m)), GRB.MINIMIZE)
14        model.addConstr(quicksum(output_prices[r] * dy[r] for
15            r in range(s)) >= A, "A_constraint")
16        for i in range(m):
17            model.addConstr(
18                quicksum(lambdas[j] * input_data[j][i] for j
19                    in range(n)) <= input_efficiency[k] *
20                    (input_data[k][i] + dx[i]),
21                f"input_constraint_{i}" )
22        for r in range(s):
23            model.addConstr(
24                quicksum(lambdas[j] * output_data[j][r] for j
25                    in range(n)) >= (dy[r] +
26                    output_data[k][r]),
27                f"output_constraint_{r}")
28        model.addConstr(quicksum(lambdas[j] for j in
29            range(n)) == 1, "sum_lambda")
30        for j in range(n):
31            model.addConstr(lambdas[j] >= 0,
32                f"non_negative_lambda_{j}")
33        model.optimize()
34        supermarket = k + 1
35        if model.status == GRB.Status.OPTIMAL:
```

```

26         dx_values = [dx[i].x for i in range(m)]
27         dy_values = [dy[r].x for r in range(s)]
28     else:
29         dx_values = [None] * m
30         dy_values = [None] * s
31     results.append([supermarket] + dx_values + dy_values)
32     columns = ['Supermarket'] + [f"dx_{i + 1}" for i in
33         range(m)] + [f"dy_{r + 1}" for r in range(s)]
34     results_df = pd.DataFrame(results, columns=columns)
35     return results_df
36
37 def budget_based_dea(input_data, output_data,
38     output_efficiency, input_prices, output_prices, B):
39     n = len(input_data)
40     m = len(input_data[0])
41     s = len(output_data[0])
42     results = []
43     for k in range(n):
44         model = Model(f"Budget_based_DEA_Supermarket_{k + 1}")
45         lambdas = model.addVars(n, name="lambda", lb=0)
46         dx = model.addVars(m, name="dx", lb=0)
47         dy = model.addVars(s, name="dy", lb=0)
48         model.setObjective(quicksum(output_prices[r] * dy[r]
49             for r in range(s)), GRB.MAXIMIZE)
50         model.addConstr(quicksum(input_prices[i] * dx[i] for
51             i in range(m)) <= B, "B_constraint")
52         for i in range(m):
53             model.addConstr(
54                 quicksum(lambdas[j] * input_data[j][i] for j
55                     in range(n)) <= (input_data[k][i] + dx[i]),
56                 f"input_constraint_{i}" )
57         for r in range(s):
58             model.addConstr(
59                 quicksum(lambdas[j] * output_data[j][r] for j
60                     in range(n)) >= output_efficiency[k] *
61                     (dy[r] + output_data[k][r]),
62                 f"output_constraint_{r}" )
63         model.addConstr(quicksum(lambdas[j] for j in
64             range(n)) == 1, "sum_lambda")
65         for j in range(n):
66             model.addConstr(lambdas[j] >= 0,

```

```

        f"non_negative_lambda_{j}")
59     model.optimize()
60     supermarket = k + 1
61     if model.status == GRB.Status.OPTIMAL:
62         dx_values = [dx[i].x for i in range(m)]
63         dy_values = [dy[r].x for r in range(s)]
64     else:
65         dx_values = [None] * m
66         dy_values = [None] * s
67     results.append([supermarket] + dx_values + dy_values)
68 columns = ['Supermarket'] + [f"dx_{i + 1}" for i in
    range(m)] + [f"dy_{r + 1}" for r in range(s)]
69 results_df = pd.DataFrame(results, columns=columns)
70 return results_df

```

### A.3 university A efficiency calculation code

Listing A.5: UniversityAefficiencyCode

```

1  import numpy as np
2  from scipy.optimize import minimize
3  def objective_function(x):
4      lambda_A, lambda_B, lambda_C, theta = x
5      return theta
6  def constraint1(x):
7      lambda_A, lambda_B, lambda_C, theta = x
8      return 1.96 * (np.sqrt(1.5 * lambda_B**2 + 1.4 *
    (lambda_A - theta)**2 + 2 * 0.9) + np.sqrt(1.2 *
    lambda_C**2 + 1.4 * (lambda_A - theta)**2 + 2 * 0.6))
    - (14 * lambda_A + 15 * lambda_B + 12 * lambda_C - 14
    * theta)
9  def constraint2(x):
10     lambda_A, lambda_B, lambda_C, theta = x
11     return 5 * theta - (5 * lambda_A + 8 * lambda_B + 7 *
    lambda_C)
12  def constraint3(x):
13     lambda_A, lambda_B, lambda_C, theta = x
14     return 9 * lambda_A + 5 * lambda_B + 4 * lambda_C - 9

```

```

15 def constraint4(x):
16     lambda_A, lambda_B, lambda_C, theta = x
17     return 4 * lambda_A + 7 * lambda_B + 9 * lambda_C - 4
18 def constraint5(x):
19     lambda_A, lambda_B, lambda_C, theta = x
20     return 16 * lambda_A + 10 * lambda_B + 13 * lambda_C - 16
21 def constraint6(x):
22     lambda_A, lambda_B, lambda_C, theta = x
23     return lambda_A + lambda_B + lambda_C - 1
24 constraints = (
25     {'type': 'ineq', 'fun': constraint1},
26     {'type': 'ineq', 'fun': constraint2},
27     {'type': 'ineq', 'fun': constraint3},
28     {'type': 'ineq', 'fun': constraint4},
29     {'type': 'ineq', 'fun': constraint5},
30     {'type': 'eq', 'fun': constraint6},)
31 bounds = [(0, None), (0, None), (0, None), (0, None)]
32 x0 = np.array([0.3, 0.3, 0.4, 0.5])
33 result = minimize(objective_function, x0,
34                   constraints=constraints, bounds=bounds, method='SLSQP')
35 print('Optimal Solution:')
36 print('Lambda_A:', result.x[0])
37 print('Lambda_B:', result.x[1])
38 print('Lambda_C:', result.x[2])
39 print('Theta:', result.x[3])

```

## A.4 Efficiency calculation Code for 20 DMUs

Listing A.6: Data in numpy array format

```

1 import numpy as np
2
3 input_means = np.array([
4     [9.131, 18.79, 7.228],..., [7.823, 17.74, 13.06]
5 ])
6
7 input_variances = np.array([
8     [0.05, 8.81, 0.58],..., [0.58, 10.1, 8.88]

```

```

9  ])
10
11  output_means = np.array([
12      [149.85, 49.621, 4.701, 4.748, 30.09], ..., [399.25,
13      40.985, 11.51, 6.432, 82.13]
14  ])
15  output_variances = np.array([
16      [48.51, 18.01, 0.41, 0.41, 24.34], ..., [10.41, 15.93,
17      0.99, 3.25, 9.891]
18  ])

```

Below is the code for the efficiency calculation of 20 banks

Listing A.7: Stochastic efficiency code

```

1
2  import gurobipy
3  from scipy.stats import norm
4  from gurobipy import Model, GRB, quicksum
5  import pandas as pd
6
7
8  def solve_stochastic_dea(input_means, input_variances,
9      output_means, output_variances, alpha, dmu_index):
10      num_DMUs, num_inputs = input_means.shape
11      _, num_outputs = output_means.shape
12
13      model = Model("Stochastic DEA")
14
15      theta = model.addVar(lb=0, name="theta")
16      lambdas = model.addVars(num_DMUs, lb=0, name="lambda")
17      p_plus = model.addVars(num_inputs, lb=0, name="p_plus")
18      p_minus = model.addVars(num_inputs, lb=0, name="p_minus")
19      q_plus = model.addVars(num_outputs, lb=0, name="q_plus")
20      q_minus = model.addVars(num_outputs, lb=0, name="q_minus")
21
22      model.setObjective(theta, GRB.MINIMIZE)
23
24      for i in range(num_inputs):
25          model.addConstr(quicksum(lambdas[j] * (input_means[j,

```

```

25         i] - norm.ppf(alpha) * np.sqrt(input_variances[j,
        i])) for j in range(num_DMUs)) <= theta *
        (input_means[dmu_index, i] - norm.ppf(alpha) *
        np.sqrt(input_variances[dmu_index, i]))
26     model.addConstr(quicksum(lambdas[j] *
27         input_variances[j, i] for j in range(num_DMUs)) -
28         theta * input_variances[dmu_index, i] == p_plus[i]
        - p_minus[i])
29
30     for r in range(num_outputs):
31         model.addConstr(quicksum(lambdas[j] *
32             (output_means[j, r] + norm.ppf(alpha) *
33             np.sqrt(output_variances[j, r])) for j in
34             range(num_DMUs)) >= output_means[dmu_index, r] +
35             norm.ppf(alpha) *
36             np.sqrt(output_variances[dmu_index, r]))
37         model.addConstr(quicksum(lambdas[j] *
38             output_variances[j, r] for j in range(num_DMUs)) -
39             output_variances[dmu_index, r] == q_plus[r] -
40             q_minus[r])
41
42     model.optimize()
43     return theta.X
44
45     alpha_levels = [0.999, 0.5, 0.2, 0.01, 0.05, 0.1, 0.001]
46     num_DMUs = len(input_means)
47
48     efficiency_results = pd.DataFrame(index=[f"DMU{i+1}" for i in
49         range(num_DMUs)], columns=alpha_levels)
50
51     for alpha in alpha_levels:
52         for dmu_index in range(num_DMUs):
53             efficiency = solve_stochastic_dea(input_means,
54                 input_variances, output_means, output_variances,
55                 alpha, dmu_index)
56             efficiency_results.at[f"DMU{dmu_index+1}", alpha] =
57                 efficiency
58
59     print("Efficiency results for each DMU with different alpha
60         levels:")
61     print(efficiency_results)

```

## A.5 Code for input level estimation in case of Stochastic data

Listing A.8: input level estimation of 20 Banks

```
1 from gurobipy import Model, GRB, quicksum
2 import numpy as np
3 from scipy.stats import norm
4
5 def dea_minimize(n, m, s, x, a, y, b,o, beta, D, k,
6     sigma_bar, theta, w,alp):
7     model = Model("DEA")
8
9     Lambda = model.addVars(n, lb=0, name="lambda")
10    t_plus = model.addVars(m, lb=0, name="t_plus")
11    t_minus = model.addVars(m, lb=0, name="t_minus")
12    p_plus = model.addVars(m, lb=0, name="p_plus")
13    p_minus = model.addVars(m, lb=0, name="p_minus")
14    q_plus = model.addVars(s, lb=0, name="q_plus")
15    q_minus = model.addVars(s, lb=0, name="q_minus")
16    alpha = model.addVars(m, lb=0, name="alpha")
17    C = model.addVars(m, lb=0, name="C")
18    objective = quicksum(w[i] * (alpha[i] + k * C[i]) for i
19        in range(m))
20    model.setObjective(objective, GRB.MINIMIZE)
21
22    phi_inv_alpha = norm.ppf(alp)
23
24    for i in range(m):
25        model.addConstr(quicksum(Lambda[j] * x[i, j] for j in
26            range(n)) - phi_inv_alpha * sigma_bar * (p_plus[i]
27            + p_minus[i]) <= theta * alpha[i])
28
29    for r in range(s):
30        model.addConstr(quicksum(Lambda[j] * y[r, j] for j in
31            range(n)) + phi_inv_alpha * sigma_bar * (q_plus[r]
32            + q_minus[r]) >= beta[r])
```



```

26     for i in range(m):
27         model.addConstr(alpha[i] - phi_inv_alpha * sigma_bar
28             * (t_plus[i] + t_minus[i]) >= x[i, o])
29     for i in range(m):
30         model.addConstr(quicksum(Lambda[j] * a[i, j] for j in
31             range(n)) - theta * C[i] == p_plus[i] - p_minus[i])
32     for r in range(s):
33         model.addConstr(quicksum(Lambda[j] * b[r, j] for j in
34             range(n)) - D[r] == q_plus[r] - q_minus[r])
35     for i in range(m):
36         model.addConstr(C[i] - a[i, o] == t_plus[i] -
37             t_minus[i])
38
39     model.optimize()
40
41     if model.status == GRB.Status.OPTIMAL:
42         print("Optimal solution found")
43         for i in range(m):
44             print(f"alpha_{i+1}o: {alpha[i]}")
45             print(f"C_{i+1}o: {C[i]}")
46     else:
47         print("No optimal solution found")

```

# Appendix B

## Data

### B.1 Hypothetical data for the three universities

#### Input Data

Table B.1: Hypothetical data for the input and outputs for the three universities [11]

University	Deterministic Inputs	Outputs		
	No. of Professors	No. of Diploma	No. of Bachelors	No. of Masters
A	5	9	4	16
B	8	5	7	10
C	7	4	9	13

#### Output data

Table B.2: Hypothetical data for the stochastic input (budget) for the three universities [11]

University	$\mu$	$\sigma^2$	Covariances
A	14	1.4	$\text{cov}(x_A, x_B) = 0.9$
B	15	1.5	$\text{cov}(x_A, x_C) = 0.6$
C	12	1.2	$\text{cov}(x_B, x_C) = 0.7$

### B.2 Data of 20 Banks

Table B.3: The inputs of 20 bank branches. [13](#)

DMU	Input 1		Input 2		Input 3	
	Mean	Variance	Mean	Variance	Mean	Variance
DMU01	9.131	0.05	18.79	8.81	7.228	0.58
DMU02	10.59	0.53	44.32	24.1	1.121	0.02
DMU03	6.712	0.86	19.73	27.7	19.21	0.47
DMU04	11.91	0.31	17.43	12.2	59.47	5.99
DMU05	7.012	0.02	10.38	2.12	12.23	0.85
DMU06	18.99	0.88	16.67	10.8	568.6	28.1
DMU07	11.16	0.01	25.46	18.6	552.8	43.2
DMU08	15.05	0.48	123.1	42.6	14.78	0.06
DMU09	8.787	0.38	36.16	38.4	361.8	23.2
DMU10	19.88	0.25	46.41	53.1	12.81	0.38
DMU11	18.92	0.17	36.88	54.5	24.43	0.01
DMU12	20.45	0.42	100.8	31.8	115.2	19.4
DMU13	12.41	0.12	20.19	10.6	78.02	24.1
DMU14	8.051	0.79	33.21	24.3	115.3	15.6
DMU15	18.48	0.92	45.36	92.6	57.52	12.8
DMU16	10.35	0.27	11.16	3.32	43.32	36.1
DMU17	9.511	0.01	31.49	38.5	173.3	3.13
DMU18	13.71	0.18	40.32	51.4	10.88	0.12
DMU19	11.69	0.26	26.44	26.2	31.22	0.05
DMU20	7.823	0.58	17.74	10.1	13.06	8.88

Table B.4: The outputs of 20 bank branches. [12](#)

DMU	Output 1		Output 2		Output 3		Output 4		Output 5	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
DMU01	149.85	48.51	49.621	18.01	4.701	0.41	4.748	0.41	30.09	24.34
DMU02	50.772	8.011	73.132	13.11	1.815	0.02	3.035	0.68	5.823	0.689
DMU03	259.91	29.59	108.04	22.56	6.016	0.01	10.06	2.98	2.721	0.4392
DMU04	137.51	21.65	44.972	13.78	4.923	1.65	4.212	3.84	63.61	33.73
DMU05	95.901	2.521	31.633	18.94	2.718	0.44	9.024	7.23	64.55	31.25
DMU06	112.58	3.562	71.958	17.05	13.19	9.15	41.89	13.2	291.3	92.52
DMU07	192.97	14.56	78.015	19.56	7.791	3.56	15.89	2.34	7.499	1.268
DMU08	724.38	66.02	219.69	37.56	35.32	15.3	23.98	10.4	361.6	48.35
DMU09	548.15	41.86	86.225	28.31	17.64	3.67	86.23	17.2	565.2	175.1
DMU10	1229.1	69.02	194.58	17.35	25.91	12.3	86.76	22.1	600.6	86.34
DMU11	11557	718.1	155.32	49.13	166.6	14.1	8.142	3.11	119.9	14.89
DMU12	1132.1	35.35	248.16	23.89	46.88	26.3	31.85	4.1	96.21	44.47
DMU13	438.39	17.41	104.41	25.71	10.68	1.04	30.22	3.77	331.9	17.16
DMU14	260.82	15.32	87.369	23.44	8.415	3.52	6.101	2.86	36.93	5.777
DMU15	11190	1214	166.44	10.69	65.12	16.1	132.7	13.9	919.1	133.1
DMU16	709.85	42.19	159.48	32.78	36.89	6.28	12.15	2.51	79.66	10.52
DMU17	308.11	43.72	107.03	29.39	11.87	1.76	13.63	1.57	342.3	134.9
DMU18	259.21	34.21	81.779	21.65	5.212	1.94	8.021	3.52	107.9	13.07
DMU19	381.33	57.38	72.993	29.26	5.165	1.79	50.32	7.47	577.2	113.4
DMU20	399.25	10.41	40.985	15.93	11.51	0.99	6.432	3.25	82.13	9.891

# Appendix C

## Models

Table C.1: Generalized Input and Output Oriented inverse DEA Models

Model	Input-Oriented	Output-Oriented
<b>Objective</b>	$\min (\Delta x_{1o}, \Delta x_{2o}, \dots, \Delta x_{mo})$	$\max (\Delta y_{1o}, \Delta y_{2o}, \dots, \Delta y_{so})$
<b>Constraints</b>	$\sum_{j=1}^n \lambda_j x_{ij} \leq \theta_0 (x_{i0} + \Delta x_{i0})$ $\sum_{j=1}^n \lambda_j y_{rj} \geq y_{r0} + \Delta y_{r0}$ $\delta_1 (\sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu) = \delta_1$ $\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n$ $\delta_i \in \{0, 1\}; i = 1, 2, 3$	$\sum_{j=1}^n \lambda_j y_{rj} \geq \phi_0 (y_{r0} + \Delta y_{r0})$ $\sum_{j=1}^n \lambda_j x_{ij} \leq x_{i0} + \Delta x_{i0}$ $\delta_1 (\sum_{j=1}^n \lambda_j + \delta_2 (-1)^{\delta_3} \nu) = \delta_1$ $\nu \geq 0, \lambda_j \geq 0, j = 1, \dots, n$ $\delta_i \in \{0, 1\}; i = 1, 2, 3$

Input-oriented BCC Model	Stochastic form of Input-oriented BCC model
Minimize $Z_p = \theta$ <b>Subject to</b> $\sum_{i=1}^n \lambda_i x_i \leq \theta x_p, (\forall j = \{1, \dots, m\})$ $\sum_{i=1}^n \lambda_i y_i \geq y_p, (\forall k = \{1, \dots, s\})$ $\sum_{i=1}^n \lambda_i = 1$ $\lambda_i \geq 0, i = 1, 2, \dots, n$	Minimize $Z_p = \theta$ <b>Subject to</b> $\sum_{i=1}^n \lambda_i \mu_i - \phi \mu_p \leq e \sqrt{\sum_{i=1, i \neq p}^n \lambda_i^2 \sigma_i^2 + (\lambda_p - \theta)^2 \sigma_p^2 + 2cov(x_i, x_p)} \quad \forall j = 1, 2, \dots, J_s$ $\sum_{i=1}^n \lambda_i x_i \leq \theta x_p, (\forall j = \{1, \dots, J_D\})$ $\sum_{i=1}^n \lambda_i y_i \geq y_p, (\forall k = \{1, \dots, s\})$ $\sum_{i=1}^n \lambda_i = 1$ $\lambda_i \geq 0, i = 1, 2, \dots, n$
Output-oriented BCC model	Stochastic form of Output-oriented BCC model
Maximize $Z_p = \phi$ <b>Subject to</b> $\sum_{i=1}^n \lambda_i x_i \leq x_p, (\forall j = \{1, \dots, m\})$ $\sum_{i=1}^n \lambda_i y_i \geq \phi y_p, (\forall k = \{1, \dots, s\})$ $\sum_{i=1}^n \lambda_i = 1$ $\lambda_i \geq 0, i = 1, 2, \dots, n$	Maximize $Z_p = \phi$ <b>Subject to</b> $\sum_{i=1}^n \lambda_i x_i \leq \theta x_p, (\forall j = \{1, \dots, m\})$ $\sum_{i=1}^n \lambda_i \mu_i - \phi \mu_p \geq e \sqrt{\sum_{i=1, i \neq p}^n \lambda_i^2 \sigma_i^2 + (\lambda_p - \phi)^2 \sigma_p^2}$ $\sum_{i=1}^n \lambda_i = 1$ $\lambda_i \geq 0, i = 1, 2, \dots, n$ where e denotes the significance level

Table C.2: Stochastic Input-Output oriented Models