

A DEA-BASED APPROACH TO RANKING  
MULTI-CRITERIA ALTERNATIVES

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CEREN TUNCER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
INDUSTRIAL ENGINEERING

AUGUST 2006

Approval of the Graduate School of Natural and Applied Sciences

---

Prof. Dr. Canan Özgen  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Prof. Dr. Çağlar Güven  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. Murat Köksalan  
Supervisor

**Examining Committee Members**

Prof. Dr. Gülser Köksal (METU, IE) \_\_\_\_\_

Prof. Dr. Murat Köksalan (METU, IE) \_\_\_\_\_

Ass. Prof. Seçil Savaşaneril (METU, IE) \_\_\_\_\_

Ass. Prof. Esra Karasakal (METU, IE) \_\_\_\_\_

Özgür Özpeynirci ( M.S.) (TÜBİTAK) \_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Ceren Tuncer

Signature :

## **ABSTRACT**

### **A DEA-BASED APPROACH TO RANKING MULTI-CRITERIA ALTERNATIVES**

Tuncer, Ceren

M.Sc., Department of Industrial Engineering

Supervisor: Prof. Dr. Murat Köksalan

August 2006, 88 pages

This thesis addresses the problem of ranking multi-criteria alternatives. A Data Envelopment Analysis (DEA)-based approach, the Method of the Area of the Efficiency Score Graph (AES) is proposed. Rather than assessing the alternatives with respect to the fixed original alternative set as done in the existing DEA-based ranking methods, AES considers the change in the efficiency scores of the alternatives while reducing the size of the alternative set. Producing a final score for each alternative that accounts for the progress of its efficiency score, AES favors alternatives that manage to improve quickly and maintain high levels of efficiency. The preferences of the Decision Maker (DM) are incorporated into the analysis in the form of weight restrictions. The utilization of the AES scores of the alternatives in an incremental clustering algorithm is also proposed. The AES Method is applied to rank MBA programs worldwide, sorting of the programs is also performed using their AES scores. Results are compared to another DEA-based ranking method.

Keywords: Ranking, data envelopment analysis, weight restrictions.

## ÖZ

### ÇOK KRİTERLİ SEÇENEKLERİN SIRALANMASINA VERİ ZARFLAMA ANALİZİ TABANLI BİR YAKLAŞIM

Tuncer, Ceren

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Murat Köksalan

Ağustos 2006, 88 sayfa

Bu tez çalışması çok kriterli seçeneklerin sıralanması problemi üzerinedir. Veri Zarflama Analizi (DEA) tabanlı bir yaklaşım olan Etkinlik Skorları Grafiğinin Alanı Metodu (AES) önerilmiştir. Seçenekleri mevcut DEA tabanlı sıralama metodlarının yaklaşımıyla sabit başlangıç seçenek kümesine göre değerlendirmek yerine, AES seçenek kümesinin küçültülmesiyle etkinlik skorlarında oluşan değişimleri göz önünde bulundurur. Her seçenek için etkinlik skorunun gelişimini gösteren son bir skor bulan AES, hızlı gelişim gösteren ve etkinlik skorlarını yükselten seçeneklerin lehine davranır. Karar vericinin tercihleri analize ağırlıkların sınırlandırılması şeklinde dahil edilir. AES skorlarının artışı bir kümeleme yönteminde kullanılması da önerilmiştir. AES Metodu MBA programlarının sıralanmasında uygulanmış, programların sınıflandırılması da AES skorları kullanılarak yapılmıştır. Sonuçlar başka bir DEA tabanlı sıralama metodunun sonuçlarıyla karşılaştırılmıştır.

Anahtar Kelimeler: Sıralama, veri zarflama analizi, ağırlık sınırlandırılması.

To My Grandmother

## **ACKNOWLEDGEMENTS**

I would like to express my gratitude to my supervisor Prof. Dr. Murat Köksalan for his support, encouragements, insight and advice throughout the research, and also my study in Industrial Engineering in general.

I would like to thank examining committee members for their valuable suggestions and contributions.

I also wish to thank METU IE Department and TÜBİTAK for supporting my academic work.

I would also like to express my deepest appreciation for the never-ending love and support of my family; nothing would be possible without you. I am also grateful to my friends who were there for me always.

## TABLE OF CONTENTS

<b>PLAGIARISM.....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>ÖZ.....</b>	<b>v</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>vii</b>
<b>TABLE OF CONTENTS .....</b>	<b>viii</b>
<b>LIST OF TABLES .....</b>	<b>ix</b>
<b>LIST OF FIGURES .....</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>xii</b>
<b>CHAPTERS:</b>	
<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. LITERATURE REVIEW .....</b>	<b>3</b>
2.1. Ranking Methods .....	3
2.2. Sorting Methods.....	4
2.3. Data Envelopment Analysis.....	5
2.3.1. The Use of Weight Restrictions in DEA.....	6
2.3.2. DEA-Based Ranking Methods.....	9
<b>3. BACKGROUND INFORMATION ON DEA MODELING, SOME WEIGHT RESTRICTION METHODS AND RANKING IN DEA .....</b>	<b>11</b>
3.1. DEA Basics and Modeling.....	11
3.2. Virtual Weight Restrictions in DEA .....	14
3.3. Weight Restrictions with Assurance Regions I and II .....	15
3.4. Cross-Efficiency Analysis .....	17
3.5. Radii of Classification Preservation in DEA .....	18
<b>4. A NEW APPROACH TO RANKING AND SORTING ALTERNATIVES IN DEA CONTEXT .....</b>	<b>20</b>
4.1. Examples and Discussions about Some Ranking Methods in DEA.....	20
4.2. Some Methods Developed to Rank DMUs in DEA .....	31
4.3. A New Approach to Ranking and Sorting DMUs in DEA.....	41
<b>5. EXPERIMENTS .....</b>	<b>49</b>
5.1. The MBA-Ranking Problem.....	49
5.2. Weight Restrictions in Ranking MBA Programs by the AES Method.....	53
5.3. The Results of the MBA Ranking Problem with the AES Method .....	62
5.4. The Results of the MBA Ranking Problem with Cross-Efficiency Method .	69
5.5. Sorting MBA Programs .....	74
<b>6. CONCLUSIONS AND FURTHER RESEARCH AREAS .....</b>	<b>80</b>
<b>REFERENCES.....</b>	<b>82</b>
<b>APPENDIX.....</b>	<b>85</b>



## LIST OF TABLES

Table 1. The Cross-Efficiency matrix.....	17
Table 2. Input/output values for the first example set .....	21
Table 3. Efficiency scores for the first example set.....	22
Table 4. Efficiency scores of the first example set after the removal of DMU 8 .....	24
Table 5. Input/output values, efficiency scores and $D_k$ values of the second example set.....	26
Table 6. The third example set in rank order of Cross-Efficiency scores.....	29
Table 7. Initial GTR Scores for the first example set .....	31
Table 8. GTR Scores for the first example set without DMU 8 .....	31
Table 9. $D_k - O_k$ values of the first example set, their best and worst possible ranks.....	32
Table 10. Ranking with optimist and pessimist approaches of $D_k - O_k$ .....	33
Table 11. The fourth example set, efficiency scores and $D_k$ values .....	35
Table 12. Efficiency scores for the first example set with the deletion of 0-11 DMUs .....	39
Table 13. The areas and efficiency scores of the first example set.....	46
Table 14. Ranking the fourth example set with the AES method.....	47
Table 15. The criteria and their weights used by FT to rank MBA Programs.....	50
Table 16. The 81 MBA Programs with their FT ranks and scores in 20 criteria.....	51
Table 17. Ranges for the virtual weight restrictions $- \mp 25\%$ and $\mp 40\%$ .....	56
Table 18. AES ranking of the first 20 programs with virtual weight restrictions ( $\mp 25\%$ ) .....	56
Table 19. AES ranking of the first 20 programs with virtual weight restrictions (+25%).....	58
Table 20. AES ranking of the first 20 programs with virtual weight restrictions (+40%).....	58
Table 21. $\mp 25\%$ and $\mp 40\%$ ranges for the weights in the AR approach.....	59
Table 22. AES ranking of the first 20 programs with AR ( $\mp 25\%$ and $\mp 40\%$ ) .....	61
Table 23. Changes in weights with virtual weight restrictions and AR (25-40% ranges).....	61
Table 24. Results of the MBA ranking problem by the AES Method.....	63

Table 25. Results of the MBA ranking problem by the Cross-Efficiency Method .....	69
Table 26. Spearman rank correlation coefficients and average rank differences .....	73
Table 27. AES sorting results of the +25% upper bounds case .....	77
Table 28. AES sorting results of the +40% upper bounds case .....	77
Table 29. Cross-Efficiency sorting results of +25% and +40 upper bounds cases .....	79
Table A. Rank lists of FT, AES, Cross-Efficiency and their differences .....	85

## LIST OF FIGURES

Figure 1. Graphical representation of the first example set.....	21
Figure 2. Graphical representation of the first example set after the removal of DMU 8.....	23
Figure 3. Graphical representation of the second example set .....	27
Figure 4. Graphical representation of the third example set.....	29
Figure 5. The general Efficiency score- $D_k$ graph .....	34
Figure 6. Efficiency Score – $D_k$ graph of the fourth example set.....	36
Figure 7. Efficiency score – $D_k$ graph of the first example set.....	37
Figure 8. Efficiency score - # of deleted DMUs graph of the first example set .....	40
Figure 9. Ranking the first example set with Efficiency score - # of deleted DMUs graph and $D_k$ scores.....	41
Figure 10. Efficiency score - # of deleted DMUs graph of 4 hypothetical DMUs...	43
Figure 11. The areas according to two measurement approaches .....	44
Figure 12. The Efficiency score - # of deleted programs graph of programs 35 and 42.....	66
Figure 13. The Efficiency score - # of deleted programs graph of programs 45 and 46.....	66
Figure 14. The Efficiency score - # of deleted programs graph of programs 47 and 48.....	67
Figure 15. The Efficiency score - # of deleted programs graph of programs 31 and 39.....	67
Figure 16. The Efficiency score - # of deleted programs graph of programs 31 and 34.....	68
Figure 17. The Efficiency score - # of deleted programs graph of programs 50 and 68.....	72
Figure 18. The Efficiency score - # of deleted programs graph of programs 17 and 17.....	72
Figure 19. The AES score distribution of the programs for +25% upper bounds ....	75
Figure 20. The AES score distribution of the programs for +40% upper bounds ....	75
Figure 21. Cross-Efficiency score distribution of the programs for +25% upper bounds.....	78
Figure 22. Cross-Efficiency score distribution of the programs for +40% upper bounds.....	78

## LIST OF ABBREVIATIONS

AES	: Area of the Efficiency Score Graph
AR	: Assurance Region
DEA	: Data Envelopment Analysis
DM	: Decision Maker
DMU	: Decision Making Unit
FT	: Financial Times
GTR	: General Tchebycheff Radius
MCDA	: Multi-Criteria Decision Aid
MCDM	: Multi-Criteria Decision Making
MPS	: Most Preferred Solution
TR	: Tchebycheff Radius
VEA	: Value Efficiency Analysis

# CHAPTER 1

## INTRODUCTION

The problem of ranking and sorting alternatives with respect to criteria the decision makers find relevant and important to their decisions has found its extensive place in both academic work and real life applications. As the existence of a single criterion in a problem of ranking/sorting alternatives is neither usual nor challenging, many methods have been proposed to deal with multi-criteria alternatives. When ranking is the issue, alternatives are ordered from the best to the worst based on their distinctive measure of performance derived. On the other hand, when sorting is the problem, alternatives are placed in groups with others that are found to be similar with respect to the assessment used. This time these groups should be ordered from the best to the worst; otherwise, that is when the groups are not ordered, the problem becomes a classification problem.

What we propose in this study is a new Data Envelopment Analysis based method to rank/sort multi-criteria alternatives. Data Envelopment Analysis (DEA) is a method which receives growing interest in the problem area of evaluating the so called “Decision Making Units” (DMUs) which consume multiple inputs to produce multiple outputs. As inputs and outputs can logically be considered as criteria, DEA is used to deal with multi-criteria alternatives. In its purest form, DEA comes up with “efficiency scores” for all DMUs which show how efficiently they can transform their inputs to outputs where each of them has great flexibility in stressing its own strengths to achieve higher scores.

Our method makes use of a new measure derived from DEA-based computations which we will argue to better represent a DMU in light of our upcoming discussions. Despite the classical DEA model and some other methods proposed upon, our method does not determine the efficiency score of alternatives based largely on their initial input-output positions relative to each other. It also presents the decision maker (DM) the opportunity to include her/his judgments about the relative importance of inputs and outputs considered.

After this introductory chapter, we review the literature in the next chapter on ranking/sorting methods, their applications of interest to us, DEA, its applications and extensions.

In chapter 3, we present the background information on DEA modeling, the inclusion of weight restrictions in DEA, relevant ranking methods based on DEA which we will use for the discussion of our method.

In chapter 4, we introduce our method with examples and discussions of justification.

In chapter 5, we provide experimental results of our method.

In the 6<sup>th</sup> and last chapter, we conclude our study and discuss further research areas.

## CHAPTER 2

### LITERATURE REVIEW

In this chapter, we review the literature on ranking and sorting techniques, Data Envelopment Analysis, its extensions, alterations and applications on ranking.

#### 2.1. Ranking Methods

In ranking problems, the alternatives in consideration are to be ranked (partially or fully) according to their scores in a number of criteria. There are several methods developed in this area. A detailed literature review of ranking methods can be found in Büyükbaşaran, 2005.

The simplest approach proposed for ranking multi-criteria alternatives is mainly about assigning fixed predetermined weights to the criteria. Each alternative's overall score is computed by multiplying its score in the criteria with the respective weights assigned and summing them up. Then, one can order the alternatives with respect to these scores. Although this method offers the ease of use and comprehension, it requires the predetermination of weights which are difficult to agree on by the parties included and which are often subject to criticism. Fixed weights give the alternatives no free space to stress their own strengths and they require strict consensus among all stakeholders. Köksalan et al. (2005) discuss more on the shortcomings of using fixed weights.

Another approach is the use of voting information on the alternatives. Methods proposed in this area rank the alternatives by comparing them in pairs or among groups by the votes they earn. Cook and Kress (1990) present a model which aggregates votes into an overall index for the purpose of representing each alternative fairly. Obata and Ishii (2003) propose a method to discriminate efficient alternatives based on voting information. Foroughi and Tamiz (2005) extend Obata and Ishii's method to include the ranking of inefficient alternatives as well. Cook (2006) studies the issue of aggregating ordinal voter preferences to a consensus achieved final ranking.

A third approach is the use of several statistical methods. These methods try to decrease the number of criteria by representing them with newly found criteria.

Multi-Criteria Decision Making (MCDM) is also a research area which contains ranking methods, some complete some partial. Most generally, MCDM is related to analyzing situations involving multi-criteria alternatives and providing the DM with assistance. An extensive review on the use of ordinal ranking methods for MCDM can be found in Landsdowne, 1996. Ordinal ranking methods require information only on the rank order of the alternatives in the criteria involved.

Büyükbaşaran (2005) proposes the method of Target-Direction-Set Value Efficiency Analysis (TDSVEA) as a ranking method. TDSVEA is an extension of Value Efficiency Analysis (VEA) which will be shortly introduced in latter sections.

Büyükbaşaran (2005) also proposes the use of a mixed integer programming (MIP) based method to rank multi-criteria alternatives.

DEA has also been used as a ranking tool in the literature. The efficiency scores found for each unit can easily be used to rank “inefficient” alternatives whose DEA scores are below 1; DEA assigns an efficiency score of 1 to all efficient alternatives. We discuss DEA-based methods with increased ranking power later.

## **2.2. Sorting Methods**

Owing to the nature of the problem and the requirement of the DMs, sometimes it is desired to rank groups of alternatives rather than rank them individually. An extensive literature review on the several methods proposed on Multi-Criteria Decision Aid (MCDA) sorting can be found in Zopounidis and Doumpos (2002).

All sorting methods involve a form of the criteria aggregation model that is developed for sorting and the methodology that is employed to define the parameters of the model.

The criteria aggregation models used in the MCDA context include the outranking relation, the utility function, discriminant functions and preference models.

The outranking relation is a binary relation that shows whether an alternative outranks another alternative. For sorting purposes, the outranking relation is used to sort alternatives by the help of heuristic assignment procedures. The most widely



used approach to sorting using outranking relation is the ELECTRE TRI Method, which constructs a preference model to the decision maker by the determination of several parameters. The ELECTRE TRI method basics and a new approach that aims to determine the mentioned parameters by the help of a non-linear optimization program is in The and Mousseau (2002).

Use of another model, the utility function, converts the scale of the criteria to utility terms. The global utility of each alternative which covers all criteria involved serves as the overall performance measure that is used to decide upon the classes. One of the methods that employ the simplest utility theory for sorting purposes is the UTADIS Method. Zopounidis and Doumpos (1999) present UTADIS basics and apply it to two real world classification problems. Besides this use of additive utility functions, more complicated utility decomposition functions, like multiplicative functions, are also used in the literature.

Use of the simple discriminant function as the criteria aggregation model requires the appropriate selection of cut-off points for the discriminant scores of the alternatives. Besides the simple discriminant function, quadratic discriminant function is also used (as in Falk and Karlov, 2001).

The above forms of sorting are based on the assumption that specific functional forms can represent the DM's preferences and it is assumed that they can represent the complexity of the DM's preferences. However, this assumption is unrealistic when the DM's preferences are highly complex and difficult to acquire. For this reason, methods that deduce preference models based on example decisions made on reference alternatives are also proposed.

The second step of sorting in MCDA context, the determination of the model's parameters, is either performed by mathematical programming techniques or inductive learning approaches depending on the form of the model.

### **2.3. Data Envelopment Analysis**

Data Envelopment Analysis has been receiving great interest since it was originally proposed by Charnes, Cooper and Rhodes (1978). It has been applied to performance, productivity and efficiency analysis of a wide range of decisional entities. Seiford (1996) reviews the evolution of DEA, the most influential

developments and novel applications. Seiford (1997) also provides an extensive bibliography for DEA starting from its first proposal.

DEA measures the efficiency scores of Decision Making Units which perform the same tasks by consuming the same inputs and producing the same outputs. The score of a DMU is basically the ratio of its weighted sum of outputs to that of inputs. Each DMU's score is required to be positive and limited up to 1; all efficient DMUs receive a score of 1 whereas inefficient alternatives' scores are positive numbers smaller than 1. The scores of the inefficient alternatives give information on the required radial improvement in their inputs and outputs in order for them to become efficient. One of DEA's most appealing properties is that it requires no predetermination of weights for inputs and outputs. In the original form of DEA, each DMU is free to choose its own weights to raise its efficiency score as high as possible, thus eliminating the arguments that the weights used favor some DMUs over others and also the difficult task of finding the appropriate weights.

Ever since the first proposal of DEA, there have been many variations, extensions and applications of it. The weights DMUs can choose are restricted in many ways, the preferences of the DM on some outputs/inputs over others are tried to be included in the formulations. Altered DEA applications are used to rank alternative units.

One extension of DEA is Value Efficiency Analysis (VEA) introduced by Halme et al. (1999). VEA is one approach that incorporates the preference information of the DM in the analysis. The DM implicitly expresses her/his preferences by choosing a (existing or hypothetical) Most Preferred Solution (MPS) which lies on the efficient frontier formed by the efficient alternatives. Then, each alternative's efficiency score is estimated by observing its position with respect to the utility function that passes through the MPS.

The DM's judgments can be included in the analysis also by restricting the weights the DMUs can choose. Weight restriction in DEA is studied by several researchers and we review these in detail in the next section.

### **2.3.1. The Use of Weight Restrictions in DEA**

One of the most questioned properties of the original DEA formulation is the full flexibility it offers to the DMUs when choosing their weights for inputs and

outputs. Even though this brings the mentioned advantages, many researchers feel the need to include the DM's judgments about the importance of individual inputs and outputs, their relative value and preferred vectors of them. When left unrestricted, DMUs that perform poorly in some criteria that might be very important to the DM can become efficient, as they can compensate it by their strong sides. Some DMUs may assign zero or very small weights to some inputs and outputs even in cases the DM finds this unacceptable. Another motive behind the use of weight restrictions is to enable stronger discrimination between the DMUs. When the weights are restricted, the number of efficient alternatives (whose efficiency scores are 1) drops as the weights that can make some DMUs efficient are no longer feasible. This brings the advantage of comparing the efficiency score of an increased number of DMUs, and also in the light of the DM's preferences.

There are several methods to include weight restrictions in DEA modeling. A detailed classification of these methods can be found in Allen et al. (1997).

The most direct way to restrict the weights is to force them to be greater and/or smaller than some predefined positive number. This is referred to as "absolute weight restrictions" in the literature. Absolute weight restrictions are very easy to apply since they require only the addition of simple constraints on the weights. However, the determination of the constants in the constraints is difficult. Dyson and Thanassoulis (1988) propose the use of regression analysis in the case of single input. They argue that in the presence of single input, the weights of the outputs can be interpreted as the amount of resource (the input) they consume per unit of output. When the regression analysis is carried out (assuming that the constant is small enough to ignore) they increase the coefficients found by some agreed upon percentage and thereby achieve the lower bounds on every output. This method does not allow the DMUs to choose weights for their outputs lower than the general consumption rates found by regression. In the case of multiple inputs, weights of a preferred DMU or the average of the weights of some selected DMUs can be used as base values.

Another way to restrict weights in DEA is the inclusion of relative ordering or the importance of the inputs/outputs. This kind of weight restrictions are called as "type I/II Assurance Regions" (AR-I/AR-II) by Thompson et al., (1990). When the

relation is between only inputs or outputs, it is of type AR-I; when the relation is also between inputs and outputs, it is of type AR-II.

Podinovski (2004) proposes that weight restrictions are useful only when they are derived from valid production trade-offs. He shows that every production trade-off can be converted to an equivalent AR-I/AR-II type restriction and argues that this method ensures that the targets of inefficient units are technologically realistic.

Golany (1988) proposes modifications in input/output values of the DMUs to provide AR-I type of restrictions in the weights. The method is suggested to be applied to situations where an input/output factor for the current period, the cumulative for the last two periods, three periods, and so on are all to be included as inputs/outputs and the DM wants to see lagging effects. When the input/output data are taken in this way, the model assigns bigger weights to inputs/outputs which are composed of fewer periods' data. In this way, AR-I type of restrictions is achieved, but without including any explicit weight restrictions. Ali et al. (1991) argue that data modifications proposed by Golany (1988) provide suitable representations of AR-I type weight restrictions for only strict (not weak) ordinal relationships. They also propose the accumulation of the weights themselves to achieve the same results.

Wong and Beasley (1990) propose a new method to restrict the weights without imposing direct restrictions on them. In their method, the proportion of a weighted input/output to the sum of all weighted inputs/outputs is restricted to be between determined percentage value ranges. This method offers the DMU the opportunity to express her/his judgments as the percentage importance of individual inputs/outputs. Unlike in the case of absolute weight restrictions, the dependence on the units of measurement in inputs/outputs is also avoided. Sarrico and Dyson (2004) propose a modified method of Wong and Beasley's arguing that their proportional weight restrictions discussed above can lead to infeasibility problems and targets and efficiency scores can no longer be interpreted as in the classical DEA. They used the nominator values of Wong and Beasley in absolute, AR-I and AR-II type of restrictions discussed above.

Halme and Korhonen (2000) propose the use of weight restrictions in VEA. They discussed the cases of selecting MPS after the imposition of weight restrictions

and vice versa. They concluded on the supplementarity of VEA and the use of weight restrictions.

### **2.3.2. DEA-based Ranking Methods**

In its classical form, DEA assigns an efficiency score of 1 to DMUs that lie on the efficient frontier and a positive number smaller than 1 to the others. These latter scores readily present performance measures to rank inefficient DMUs but cannot discriminate among the efficient ones. There are several methods proposed in the DEA context to rank DMUs; some to deal with discriminating among efficient alternatives, some to bring new measures of ranking other than the classical DEA score. Reviews on ranking methods in DEA context can be found in Adler et al. (2002) and Sarkis (2000).

Andersen and Petersen (1993) propose an alteration to the original DEA formulation to discriminate among the efficient units. As mentioned, the original DEA formulation limits every DMU's efficiency score up to 1. Andersen and Petersen remove this constraint for each DMU in its formulation while preserving it for other DMUs. The result is efficient units with scores greater than 1 and inefficient units less than 1 which can be used for full ranking of all DMUs. This method is sometimes referred to as the Super-Efficiency Method.

Sarkis (2000) proposes the use of AR-I weight restrictions with Super-Efficiency to include the judgments of the DM in the analysis. These weight restrictions are derived from lower and upper bounds for the importance of inputs/outputs expressed by the DM in the form of proportions.

Doyle and Green's (1994) Cross-Efficiency Method is another approach that can discriminate between all DMUs. The basic motive for this method is our prior discussion of the total flexibility of DEA in assigning weights. To avoid the assessment of a DMU only with its own selected weights, which the DMs may not find appropriate, Cross-Efficiency finds different scores of each DMU with every DMU's weights. Then, the average score computed for each DMU is accepted as the true efficiency score of it. With this method, a DMU needs to perform well with the weights of its peers as well as its own. There are cases when the weights that maximize the score of a DMU are not unique. Doyle and Green propose two formulations for choosing among these weights: aggressive and benevolent

formulations. The first chooses among its set of optimal weights so as to minimize the scores of the other DMUs and the second chooses to maximize them.

Rousseau and Semple (1995) study the sensitivity and robustness of efficiency classifications. They develop formulations that find the minimum required change (absolute or relative) in a DMU's inputs and outputs to change its classification using Tchebycheff distance. Efficient DMUs receive positive scores and inefficient DMUs receive negative scores. For an efficient DMU, the higher its score the more change is needed for it to become inefficient, thus the higher it would be placed in ranking. For an inefficient DMU, the lower its score, the bigger the improvement should be for it to become efficient and therefore the lower it would be placed in ranking.

Three different measures to rank DMUs are proposed by Sinuany-Stern et al. (1994). The first measure is the number of times a DMU becomes efficient in the solutions of other DMUs, that is, with their optimal weights for inputs and outputs. The higher this number, the higher the DMU will be ranked. The second measure is the number of DMUs that become efficient in the solution of a DMU; the higher this number, the lower the DMU will be ranked. The final measure is the number of DMUs that should leave the set for a DMU to become efficient. The higher this number, the lower the DMU will be ranked.

Another method of ranking in DEA is the use of statistical techniques in alliance with DEA. Given the division of DMUs into efficient and inefficient sets, Sinuany-Stern et al. (1994) use linear discriminant analysis to find scores for inefficient as well as efficient alternatives which can be used for ranking.

We will propose a method that we will argue to have increased ranking power as opposed to classical efficiency scores. Also, we will show that it avoids the shortcomings of the thirdly mentioned work of Sinuany-Stern et al. (1994), and also the works of Doyle and Green and also Rousseau and Semple (1995)

## **CHAPTER 3**

### **BACKGROUND INFORMATION ON DEA MODELING, SOME WEIGHT RESTRICTION METHODS AND RANKING IN DEA**

#### **3.1. DEA Basics and Modeling**

As discussed briefly, DEA is a method developed for the computation of efficiency scores of alternatives which perform the same task(s). Unlike the parametric approaches developed to handle the same job, DEA requires no assumptions about the functional form of the technology. Having only the observed data, DEA constructs an empirical production frontier and assesses the efficiency scores of the alternatives (called as Decision Making Units in DEA terminology) with respect to it. Unlike regression-based approaches, DEA study is not about the general tendencies; it is about efficient frontiers formed by alternatives that are preferable to others.

DMUs in a DEA study consume varying amounts of the same inputs to produce varying amounts of the same outputs. The input and output values are required to be non-negative; and at least one input and one output value of every DMU is required to be non-zero.

If there were one output and one input, the efficiency score of a DMU would simply be the ratio of its output to its input. In the presence of multiple inputs and outputs, DEA score is the ratio of a DMU's 'virtual' output to its 'virtual' input. These virtual input and output are determined for each DMU by the weighted sum of its inputs and outputs. In its original model, DEA imposes no restrictions on the weights other than the non-negativity constraints.

Besides the first model by Charnes, Cooper and Rhodes (1978), which is called the CCR model, there are several related DEA models in the literature. One can see Cooper, Seiford, Tone (2000) for different DEA models including CCR, BCC (by Banker, Charnes and Cooper), Additive DEA models and Slack-Based

Measurement (SBM) models. Seiford and Thrall (1990) also provide a review on CCR and BCC models.

What we will employ in this study is the ‘input-oriented’ CCR model. The CCR model accepts constant returns to scale and when it is input oriented, its objective takes the form of maximizing weighted outputs given the level of inputs.

Before we provide the mathematical model, let

indices:

n: the number of DMUs in the study,

m: the number of inputs considered,

s: the number of outputs considered,

parameters:

$y_{ik}$ : the value of output  $i$  for DMU  $k$  ( $i=1, \dots, s$  ;  $k=1, \dots, n$ ),

$x_{jk}$ : the value of input  $j$  for DMU  $k$  ( $j=1, \dots, m$  ;  $k=1, \dots, n$ ),

decision variables:

$u_i$ : the weight for output  $i$  ( $i=1, \dots, s$ ),

$v_j$ : the weight for input  $j$  ( $j=1, \dots, m$ ).

Now, our objective of maximizing the ratio of virtual output to virtual input for a DMU 0 can be expressed as (the DMU in consideration is referred to as DMU 0 in DEA models throughout the study):

$$\text{Max } \sum_{i=1}^s u_i y_{i0} / \sum_{j=1}^m v_j x_{j0}$$

Obviously, without constraints, the value for the above term is unbounded. To normalize the efficiency scores, the ratio of the virtual output to that of virtual input is restricted to be smaller than 1. Then the model becomes,

$$\text{Max } \sum_{i=1}^s u_i y_{i0} / \sum_{j=1}^m v_j x_{j0}$$

$$\sum_{i=1}^s u_i y_{ik} / \sum_{j=1}^m v_j x_{jk} \leq 1 \quad \text{for } k=1, \dots, n,$$

$$u_i, v_j \geq 0 \quad i=1, \dots, s ; j=1, \dots, m$$



The conversion of this model to a linear model requires the setting of weighted inputs of DMU 0 to a constant (which is usually 1) and maximizing weighted outputs. The general linear model then is,

$$\text{Max } Z = \sum_{i=1}^s u_i y_{i0}$$

s.t.

$$\sum_{j=1}^m v_j x_{j0} = 1$$

$$\sum_{i=1}^s u_i y_{ik} \leq \sum_{j=1}^m v_j x_{jk} \quad \text{for } k=1, \dots, n,$$

$$u_i, v_j \geq 0 \quad i=1, \dots, s ; j=1, \dots, m$$

This is the basic CCR model we will develop our method on. Note that a separate model is solved for each DMU to find its score. In each model, a single set of weights is present. DMU 0 can choose its weights in order to receive a score as high as possible. However, it should make its virtual input 1 and also virtual outputs of the other DMUs with these weights should not exceed their virtual inputs. Efficient DMUs which succeed to be on the frontier earn scores of 1, whereas inefficient DMUs' scores range from very small positive numbers to 1.

If we take the dual of the above model, we achieve the alternative form of the CCR model which is also frequently used:

Min  $\theta$

s.t.

$$\theta x_0 - X \lambda \geq 0$$

$$Y \lambda \geq y_0$$

$$\lambda \geq 0$$

where  $X \in \mathfrak{R}_+^{m \times n}$  and  $Y \in \mathfrak{R}_+^{s \times n}$  are the input and output matrices of all DMUs, and  $x_k$  and  $y_k$  are the column vectors of the inputs and outputs of DMU  $k$ .

As mentioned, these are the models for the input-oriented case. In the output oriented case of the primal CCR model, virtual output of DMU 0 is equated to 1 and its virtual input is minimized in the objective function. The model is again solved for

each DMU. The input and output oriented models of CCR result in the same efficiency scores for the DMUs. We utilize the input orientation for our study due to the characteristics of our example problem.

### 3.2. Virtual Weight Restrictions in DEA

We have reviewed weight restriction techniques in DEA in the preceding sections. One of the discussed methods is the one by Wong and Beasley (1990). We study this method further in this section since we will employ it in our approach.

The weighted sum of the inputs/outputs of a DMU is called the virtual input/output as we have called it in the previous section. Instead of restricting weights directly, Wong and Beasley propose the restriction of weighted inputs/outputs, thereby making their method referred to as virtual weight restrictions.

Virtual weight restrictions add constraints to the CCR model in the form of ranges for the ratios of individual weighted input/output values to weighted sum of all inputs/outputs. The DM specifies the minimum and maximum percentage values for the importance of the inputs/outputs that s/he wishes to impose restrictions on.

The ranges specified are expressed as:

$$a_r \leq v_r x_{rk} / \sum_{j=1}^m v_j x_{jk} \leq b_r \quad \text{for input } r \text{ and DMU } k,$$

$$a_p \leq u_p y_{pk} / \sum_{i=1}^s u_i y_{ik} \leq b_p \quad \text{for output } p \text{ and DMU } k,$$

where  $a_r$  ( $a_p$ ) and  $b_r$  ( $b_p$ ) represent the lower and upper bounds, respectively, for the inputs (outputs).

There are three ways to incorporate virtual weight restrictions into the DEA model:

1. Add restrictions of the form above for only the DMU in evaluation, DMU 0. So, the above restrictions will be imposed for  $k = 0$ .
2. Add them for all DMUs in every DMU's formulation. In this way, the proportion constraints should be satisfied by every DMU in the set. The restrictions above will be replicated for all  $k = 1, \dots, n$ .

3. Add them for DMU 0 and also the average artificial DMU. This artificial DMU is not added to the alternative set; its average values for the inputs and outputs are used only for the restrictions. This third type of use requires the insertion of the above restrictions for DMU 0 and additionally the general restrictions below:

$$a_r \leq [v_r (\sum_{k=1}^n x_{rk}/n)] / [\sum_{j=1}^m v_j (\sum_{k=1}^n x_{jk}/n)] \leq b_r \quad \text{for input } r,$$

$$a_p \leq [u_p (\sum_{k=1}^n y_{pk}/n)] / [\sum_{i=1}^s u_i (\sum_{k=1}^n y_{ik}/n)] \leq b_p \quad \text{for output } p,$$

where  $a_r$  ( $a_p$ ) and  $b_r$  ( $b_p$ ) represent the same upper and lower bounds.

The choice on the use of virtual weight restrictions depends on the degree of strictness the analyzer wishes to restrict the weights with and the computational effort s/he is willing to spend. Note that each input/output restriction for each DMU's values requires 2 additional constraints.

Again based on the idea of virtual weight restrictions, Sarrico and Dyson (2004) propose a modified version of the above-discussed method. They show that the method by Wong and Beasley can lead to infeasibilities unless the ranges are set considering the input/output values of the DMUs. Also, they argue that it may impose inappropriate restrictions on some DMUs because of others' values in inputs/outputs. They propose the use of virtual inputs/outputs with Assurance Regions which will be discussed in the next section. It is sufficient here to note that the basic point of their method is the use of weighted inputs/outputs instead of direct weights in the Assurance Region methods.

### 3.3. Weight Restrictions with Assurance Regions I and II

Thompson et al. (1990) propose the concept of Assurance Regions to assess DMUs with respect to overall efficiency rather than technical efficiency as done in classical unrestricted DEA.

An Assurance Region (AR) is a subset of all weight vectors in DEA which contains the reasonable and desired weight vectors for inputs and outputs. All

weight vectors that lie outside a defined AR are not acceptable to the DM. As a consequence, for a DMU to be efficient in such a setting, the intersection of the set of weights that can make its score 1 and the AR defined should not be empty.

There are two types of AR introduced in Thompson et al.:

1. Assurance Regions of type I: AR-I is used to include the ordering or the relative values of the inputs or the outputs in the study. One may just want to ensure that one input/output has a greater weight than the other or s/he may have relative values in mind about these weights. AR-I is of the form:

$$c_i v_k \leq v_i \leq d_i v_k \quad k = \text{the reference input, } i = \text{all inputs except } k$$

(for inputs),

$$e_j u_m \leq u_j \leq f_j u_m \quad m = \text{the reference output, } j = \text{all outputs except } m$$

(for outputs),

where  $c_i$  ( $e_j$ ) and  $d_i$  ( $f_j$ ) are the lower and upper bounds respectively for the relative values of input (output) weights with respect to the weight of the reference input (output). So, by relating each weight to the reference weight, one can relate the values of all weights to each other.

2. Assurance Regions of type II: AR-II contains AR-I, but it also includes relations between inputs and outputs. This linkage between input and output weights first of all enables the use of a single reference weight in the AR analysis. Also, absolute profitability considerations can be included in the analysis by this linkage. So, AR-II is the union of AR-I and the set of constraints below:

$$p_i u_n \leq v_i \leq r_i u_n \quad n = \text{the reference output, } i = \text{all inputs}$$

(for the case of reference output),

$$g_j v_l \leq u_j \leq h_j v_l \quad l = \text{the reference input, } j = \text{all outputs,}$$

(for the case of reference input),

where  $p_i$  ( $g_j$ ) and  $r_i$  ( $h_j$ ) are the lower and upper bounds respectively for the relative values of input (output) weights with respect to the weight of the reference output (input).

In a study with Assurance Regions, one can tighten the ranges in consecutive computations until the desired number of efficient DMUs is left. Especially in cases

of selection among the alternatives, tighter formulations help to reduce the number of eligible DMUs in light of the DM's specifications.

### 3.4. Cross-Efficiency Analysis

As mentioned in the previous chapter, there are several methods developed for ranking DMUs in DEA. The original efficiency scores can readily be used for this job; but authors like Doyle and Green avoid letting the DMUs to have absolute freedom in choosing their weights to maximize their scores. In their 1994 paper, Doyle and Green introduce the use of cross-efficiency scores to rank DMUs, both efficient and inefficient. They propose that DMUs should not be assessed only with their own chosen weights, but also with the weights of the other DMUs. They see this method as a means of 'peer appraisal' where each DMU is appraised with respect to all DMUs' strengths instead of just its.

To provide the method, let  $v_{ij}$  be the weight DMU  $j$  chooses for input  $i$ , and  $u_{kj}$  be the weight it chooses for output  $k$ . In the presence of  $m$  inputs and  $s$  outputs, the optimal set of weights for DMU  $j$  will be  $W_j = (v_{1j}, \dots, v_{mj}; u_{1j}, \dots, u_{sj})$ . These optimal weights are found by using the classical DEA model. Let  $Eff_{jr}$  be the efficiency score DMU  $j$  will earn using the optimal set of weights for DMU  $r$ .  $Eff_{jj}$  is the classical DEA score. Then, we can construct Table 1.

**Table 1. The Cross-Efficiency matrix**

		Assessed DMU				
		1	2	3	...	n
With the weights of	1	$Eff_{11}$	$Eff_{21}$	$Eff_{31}$		$Eff_{n1}$
	2	$Eff_{12}$	$Eff_{22}$	$Eff_{32}$		$Eff_{n2}$
	3	$Eff_{13}$	$Eff_{23}$	$Eff_{33}$		$Eff_{n3}$
	⋮					
	n	$Eff_{1n}$	$Eff_{2n}$	$Eff_{3n}$		$Eff_{nn}$
Average		$e_1$	$e_2$	$e_3$		$e_n$

Each column shows the efficiency scores of a DMU with every DMU's weights and the last row shows the average of these scores. These  $e_j$  values constitute the new efficiency scores which will be used for ranking.

In DEA computations, there can be cases where the optimal set of weights for a DMU is not unique. Doyle and Green propose two approaches to choose among these sets for use in Cross-Efficiency Method. The first, the aggressive approach, chooses with the objective of minimizing the sum of all other DMUs' scores with these weights. The second approach, the benevolent one, does the opposite by maximizing the sum.

The weights used in classical Cross-Efficiency Analysis are unrestricted. However, for every case that necessitates the use of weight restrictions in DEA, Cross-Efficiency can also be adjusted to include these. The only difference to the method will be the addition of the constraints on the weights to the models.

### 3.5. Radii of Classification Preservation in DEA

Rousseau and Semple (1995) propose a new method on DEA sensitivity analysis. Troubled by the fact that DEA calculations are made on only one observation and there is always the probability that observed data may be erroneous, they study the sensitivity of DEA classifications to changes in input/output values. This sensitivity analysis is used as a ranking method in Sarkis (2000).

The classical DEA scores of DMUs split them into two classes of efficient and inefficient units. Rousseau and Semple develop formulations to find how much change is needed in their inputs/outputs to reverse their classes. Their linear program TR (Tchebycheff Radius) finds the minimum absolute change needed in inputs/outputs to alter the classes, and GTR (General Tchebycheff Radius) finds the minimum relative change. The change in both formulations is in Tchebycheff distance. Let us first provide the models and then discuss the applications.

$$\begin{aligned}
 & \text{Min } \alpha^+ - \alpha^- \\
 & \text{s.t.} \\
 \text{(GTR)} \quad & Y^{(k)}\lambda - s^+ + \alpha^+ y_k - \alpha^- y_k = y_k \\
 & X^{(k)}\lambda + s^- - \alpha^+ x_k + \alpha^- x_k = x_k \\
 & e^T \lambda = 1 \\
 & \lambda, s^+, s^-, \alpha^+, \alpha^- \geq 0
 \end{aligned}$$

where  $k$  is the DMU in consideration,  $Y^{(k)}$  and  $X^{(k)}$  are the output and input matrices of all DMUs except  $k$ ,  $x_k$  and  $y_k$  are the input and output vectors of DMU  $k$ , and  $e$  is a vector of 1s. This model finds the minimum percentage change needed in all inputs and outputs of DMU  $k$  to reverse its classification. The difference of TR from this model is that it finds the absolute change needed:

$$\begin{aligned}
 & \text{Min } \alpha^+ - \alpha^- \\
 & \text{s.t.} \\
 \text{(TR)} \quad & Y^{(k)}\lambda - s^+ + \alpha^+ e_s - \alpha^- e_s = y_k \\
 & X^{(k)}\lambda + s^- - \alpha^+ e_m + \alpha^- e_m = x_k \\
 & e^T \lambda = 1 \\
 & \lambda, s^+, s^-, \alpha^+, \alpha^- \geq 0
 \end{aligned}$$

These formulations are for both efficient and inefficient DMUs; the optimal value is positive for efficient DMUs and negative for inefficient ones. For an efficient DMU, the higher the objective function value, the more reduction is needed (absolute or relative) in inputs/outputs for it to become inefficient, and in a way the ‘safer’ its position is. For an inefficient DMU, the lower the value (negative), the more improvement is needed for it to become efficient and the worse its position is. So, the results of this sensitivity analysis present a common measure of ranking for both efficient and inefficient DMUs. One just needs to order them from the highest to the lowest value. When choosing TR or GTR, it is useful to take the ‘sizes’ of the DMUs into consideration. The absolute changes needed in small sized DMUs will be relatively low even in cases they represent a large relative change.

## **CHAPTER 4**

### **A NEW APPROACH TO RANKING AND SORTING ALTERNATIVES IN DEA CONTEXT**

In this chapter, first we look at some existing ranking methods in DEA. We discuss them with examples and point out their shortcomings. Then, we look at new approaches we considered in the process of developing our final approach. Finally, we present our method with its reasoning, illustrations and modeling.

#### **4.1. Examples and Discussions about Some Ranking Methods in DEA**

##### **Ranking DMUs with the Classical DEA Score**

We have discussed DEA basics and presented its model in previous chapters. DEA assigns a positive efficiency score up to 1 to each DMU in the alternative set. The efficient frontier is formed by DMUs which have efficiency scores of 1; and all other DMUs earn scores with respect to their standing relative to the efficient frontier.

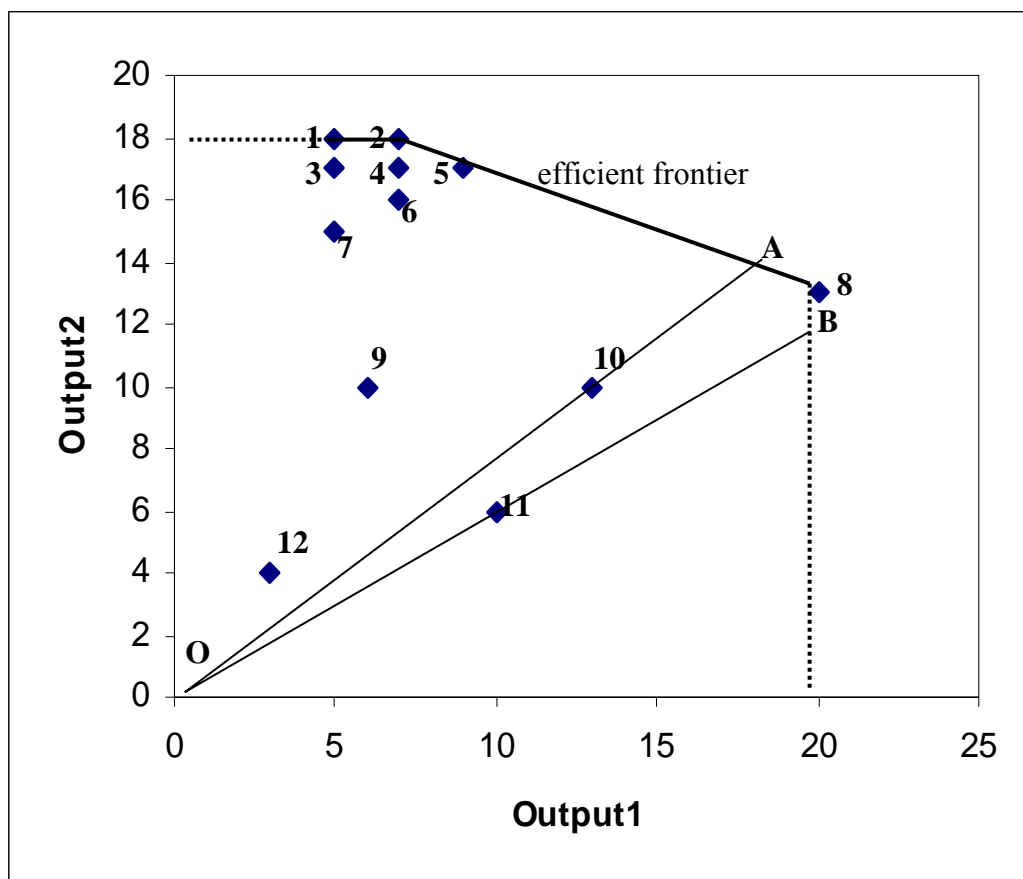
To illustrate the efficient frontier and the DMUs of a set graphically, let us use a simple example of 12 DMUs with only one input and two outputs. Assume that all DMUs' input value is 1; in cases of a single input, this condition can be achieved by adjusting the outputs of all DMUs by the division factors that make their inputs 1. Table 2 shows the input/output values of the 12 DMUs which form our first example set.

As the input value of all DMUs is 1, we can graph the DMUs by their two output values. We show each output in one axis, the choice is arbitrary. Here we show output 1 in the x-axis and 2 in the y-axis. Then, we can graph the DMUs as in Figure 1.



**Table 2. Input/output values for the first example set**

DMU #	Input	Output 1	Output 2
1	1	5	18
2	1	7	18
3	1	5	17
4	1	7	17
5	1	9	17
6	1	7	16
7	1	5	15
8	1	20	13
9	1	6	10
10	1	13	10
11	1	10	6
12	1	3	4



**Figure 1. Graphical representation of the first example set**

In this example, the efficient frontier is formed by DMUs 1, 2 and 8 which have efficiency scores of 1. The solid line represents the part of the efficient frontier formed by the convex combination of efficient DMUs. The dashed line shows the continuation of this former part which helps to ‘envelop’ the DMUs and which will be used for the radial projection of some DMUs when computing their efficiency scores. The radial projection of an inefficient DMU corresponds here to improving its output values proportionally until the boundary of the efficient frontier is reached. It should be noted here that the efficient frontier in Figure 1 is valid for the “constant returns to scale” assuming CCR model which we employ in our method. The reference point reached on the frontier by proportional improvement of an inefficient DMU is considered as a target point for it and therefore used for the computation of its efficiency score.

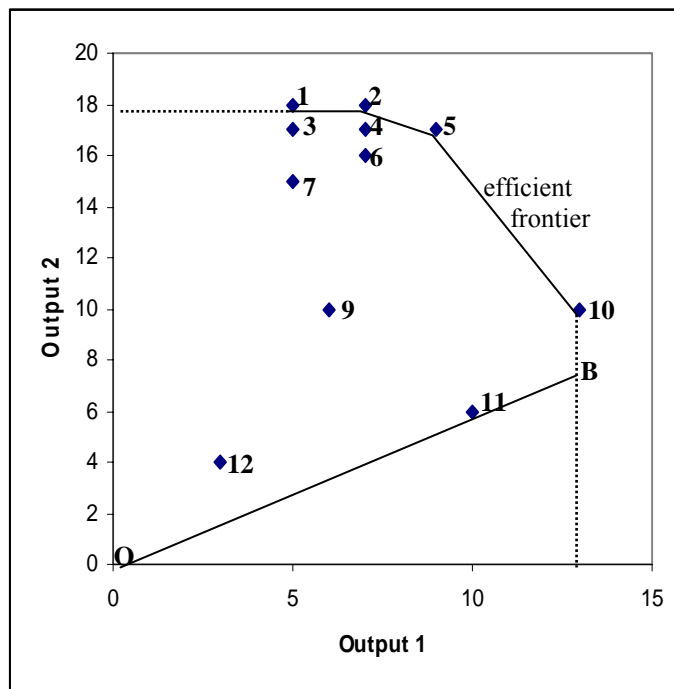
The efficiency score of an inefficient DMU is the ratio of its Euclidean distance from the origin, to the distance from the origin to its radial projection point on the frontier. Let  $|OX|$  denote the Euclidean distance from point O to point X. Then, the efficiency score of DMU 10 is  $|O10| / |OA|$  and the score of DMU 11 is  $|O11| / |OB|$ . The efficiency scores of all DMUs are provided in Table 3.

**Table 3. Efficiency scores for the first example set**

DMU #	Efficiency Score
1	1.0000
2	1.0000
3	0.9444
4	0.9516
5	0.9888
6	0.9033
7	0.8333
8	1.0000
9	0.5947
10	0.7249
11	0.5000
12	0.2490

When we observe Figure 1, we can see that DMU 8 is an outlier which by itself affects the efficient frontier substantially. Let us consider the two DMUs, 10

and 11, again. In the absence of 8, it is clear that these DMUs will perform significantly better. If we remove only DMU 8 from the set, the resulting graph and efficiency scores will be as in Figure 2 and Table 4. Before the removal of DMU 8, 10 was one of the worst four DMUs in the set; after the removal, its efficiency score becomes 1. DMU 11 has also greatly improved its efficiency score from 0.5000 to 0.7692.



**Figure 2. Graphical representation of the first example set after the removal of DMU 8**

As illustrated with this example, the classical DEA score is totally dependent on the position of the DMUs with respect to the efficient frontier formed by efficient DMUs. This frontier may consist of outlier DMUs which should not be taken as a basis for the performances of other DMUs. The removal of even one DMU can change the efficiency scores considerably, which also leads to serious changes in the relative standing of the DMUs with respect to each other in the ranking list. Therefore, the classical DEA scores should not be the only criterion for ranking alternatives.

**Table 4. Efficiency scores of the first example set after the removal of DMU 8**

<b>DMU #</b>	<b>Efficiency Score</b>
1	1.0000
2	1.0000
3	0.9444
4	0.9534
5	1.0000
6	0.9069
7	0.8333
9	0.6259
10	1.0000
11	0.7692
12	0.2824

**Ranking DMUs with the Number of DMUs That Should Leave the Set for Each of Them to Become Efficient**

Sinuany-Stern et al. (1994) propose a measure to rank inefficient DMUs. This measure counts the minimum number of DMUs that should leave the set for each DMU in the set to be efficient. We will refer to this measure as ‘ $D_k$ ’ (deleted for DMU  $k$ ) where  $k$  denotes the DMU under consideration. Obviously,  $D_k$  of all efficient DMUs are 0; for inefficient DMUs, the lower this measure is, the higher they would be ranked.

The  $D_k$  measure is an indicator of preference; all DMUs that should be deleted for the efficiency of a DMU are likely to be preferred. The deleted DMUs can be efficient or inefficient depending on the position of the DMU under consideration. The reason for a DMU to leave the set for the efficiency of another is that its efficiency cannot stay at or below the level of 1 with the weights that make the other’s efficiency 1. So, the model to measure  $D_k$  should allow some DMUs to receive scores above 1, but it also should count them and keep them to a minimum while ensuring the DMU under consideration becomes efficient. We develop a CCR-based model to find the  $D_k$  values. It should be modeled and solved for each DMU and its optimum objective function value gives the  $D_k$  measure. Our model is presented next.

$$\begin{aligned}
D_k &= \text{Min } Z = \sum_{k=1}^n N_k \\
&\text{s.t.} \\
(P_k) \quad &\sum_{j=1}^m v_j x_{j0} = 1 \\
&\sum_{i=1}^s u_i y_{i0} = 1 \\
&\sum_{i=1}^s u_i y_{ik} \leq \sum_{j=1}^m v_j x_{jk} + N_k * M \quad \text{for all } k \text{ except } 0 \\
&N_k \in \{0,1\} \\
&u_i, v_j \geq 0 \quad i=1,\dots,s ; j=1,\dots,m.
\end{aligned}$$

where  $M$  is a sufficiently large constant that allows the weighted outputs of DMUs to become as high as they desire.

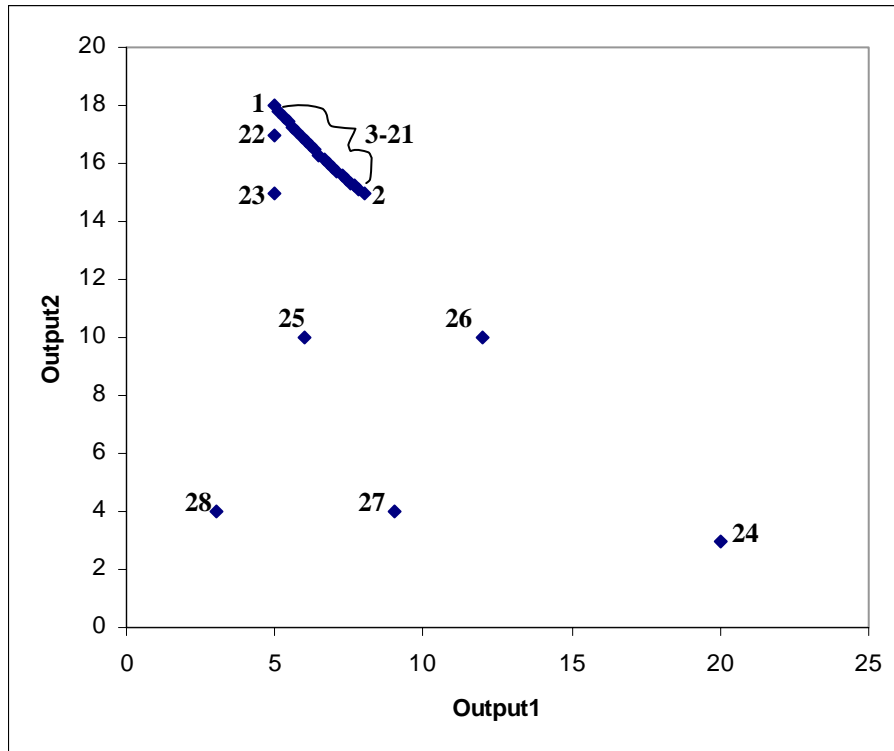
The use of  $D_k$  as a measure of ranking is meaningful in the sense that it shows the number of DMUs that are likely to be preferred to a given unit. It seems logical that a DMU that needs fewer rivals to leave the set for its efficiency should be ranked higher than one that needs more. However, patterns in the DMU set affect the  $D_k$  distribution greatly even in cases it is unreasonable. There may be DMUs that have very close input/output values. Since their part of the plot is crowded, DMUs that are somewhat farther from the efficient frontier than others will have higher  $D_k$  values even though there exists no significant difference in their performances in inputs/outputs. On the other hand, a poor performer may receive a lower  $D_k$  value due to the characteristics of the set and hence be positioned higher in the ranking list.

Let us illustrate our argument with a new example, again with two outputs and one input, which is 1 for all DMUs. The plot of this second example set is presented in Figure 3.

The efficient DMUs in this second example set are 1, 2 and 24. Although not clearly noticeable due to their very close output values, there are 19 DMUs lying between 1 and 2 that fall slightly inside the efficient frontier. Table 5 shows the output values of the DMUs, their efficiency scores and  $D_k$  values.

**Table 5. Input/output values, efficiency scores and  $D_k$  values of the second example set**

<b>DMU #</b>	<b>Input</b>	<b>Output 1</b>	<b>Output 2</b>	<b>Efficiency Score</b>	<b><math>D_k</math></b>
1	1	5.00	18.00	1.0000	0
2	1	8.00	15.00	1.0000	0
3	1	5.15	17.79	0.9972	1
4	1	5.30	17.60	0.9956	2
5	1	5.45	17.42	0.9944	3
6	1	5.60	17.25	0.9935	4
7	1	5.75	17.08	0.9928	5
8	1	5.90	16.92	0.9922	6
9	1	6.05	16.76	0.9919	7
10	1	6.20	16.61	0.9916	8
11	1	6.35	16.46	0.9915	9
12	1	6.50	16.31	0.9915	10
13	1	6.65	16.16	0.9917	10
14	1	6.80	16.01	0.9919	9
15	1	6.95	15.87	0.9923	8
16	1	7.10	15.74	0.9928	7
17	1	7.25	15.60	0.9935	6
18	1	7.40	15.47	0.9943	5
19	1	7.55	15.34	0.9952	4
20	1	7.70	15.22	0.9964	3
21	1	7.85	15.10	0.9979	2
22	1	5.00	17.00	0.9565	6
23	1	5.00	15.00	0.8696	21
24	1	20.00	3.00	1.0000	0
25	1	6.00	10.00	0.6957	17
26	1	12.00	10.00	0.9565	1
27	1	9.00	4.00	0.5652	2
28	1	3.00	4.00	0.3043	25



**Figure 3. Graphical representation of the second example set**

When we look at DMUs 1 to 21, we can see that their output values and especially efficiency scores are very close to each other. However, their  $D_k$  values range widely from 0 to 10. We placed DMUs 3-21 between 1 and 2 in such a way that their  $D_k$  values increase as we move towards their midpoint. This is an example created to illustrate the case and therefore somewhat extreme, but any crowded region in a problem setting will cause problems of the same sort even if not so severe.

### **Ranking DMUs with Cross-Efficiency Scores**

The Cross-Efficiency Method is introduced in the previous chapters. This method is based on evaluation of the DMUs not only with their own optimal weights, but also with the weights of all other DMUs in the set. This approach can be useful in cases when there are DMUs that have undesirable levels of some inputs/outputs but can still achieve high efficiency scores by stressing their strengths to their limit. With Cross-Efficiency Method, since each DMU is assessed also with other DMUs' strengths, this problem is prevented to some extent. However, this

property may also bring its own problems. Assume that all DMUs in the set have acceptable levels of inputs/outputs, but most of the DMUs are gathered together because they have close levels of inputs/outputs. In this case, these DMUs will naturally have similar optimal weights and they will receive high scores with other units' weights as well as their own. On the other hand, DMUs that perform differently than this majority will receive lower efficiency scores when assessed with their weights.

Let us introduce here a new example set of DMUs and apply Cross-Efficiency Analysis for ranking. Figure 4 illustrates the 12 DMUs of this third example set, they again have two outputs and a single input of 1. Table 6 shows the DMUs in the rank order of their Cross-Efficiency scores; it also includes their output values.

As apparent in Figure 4, there is crowding in the area consisting of DMUs 4, 5, 6, 7, 8, 9, 10 and 12. On the other hand, the remaining DMUs, including DMUs 1 and 2 that are on the efficient frontier, do not have close neighbors. When we look at the ranking results of Cross-Efficiency Analysis in Table 6, we see that DMU 7 ranks the first and all DMUs in the crowded region follow it uninterruptedly except for DMU 12, which lets DMU 2 to get ahead of it. Only then DMU 1 can find a place in the tenth rank. This under ranking of DMUs 1 and 2 is the result of crowd-favoring Cross-Efficiency scores. DMUs 1 and 2 do not have unacceptably low levels of outputs; in fact, DMU 1's values are just the reverse of the ones of DMU 7 which ranks the first. If DMU 1 instead of 7 had close neighbors, it would be ranked much higher with no change in performance. It is clear that if we keep DMUs 1 and 7 fixed and reverse the output values of the remaining DMUs, they will just change ranks.

The use of Cross-Efficiency scores is appropriate in cases where there is no significant crowding in certain areas or when the DMUs falling apart have undesirable input/output levels so that ranking them low is the consent of the DM. Otherwise, Cross-Efficiency Analysis will penalize the DMUs that just perform different from the majority, which is against the basic reasoning of DEA. As we mentioned before, DEA is not about central tendencies, it is about stressing your strengths within the limits imposed by your rivals.



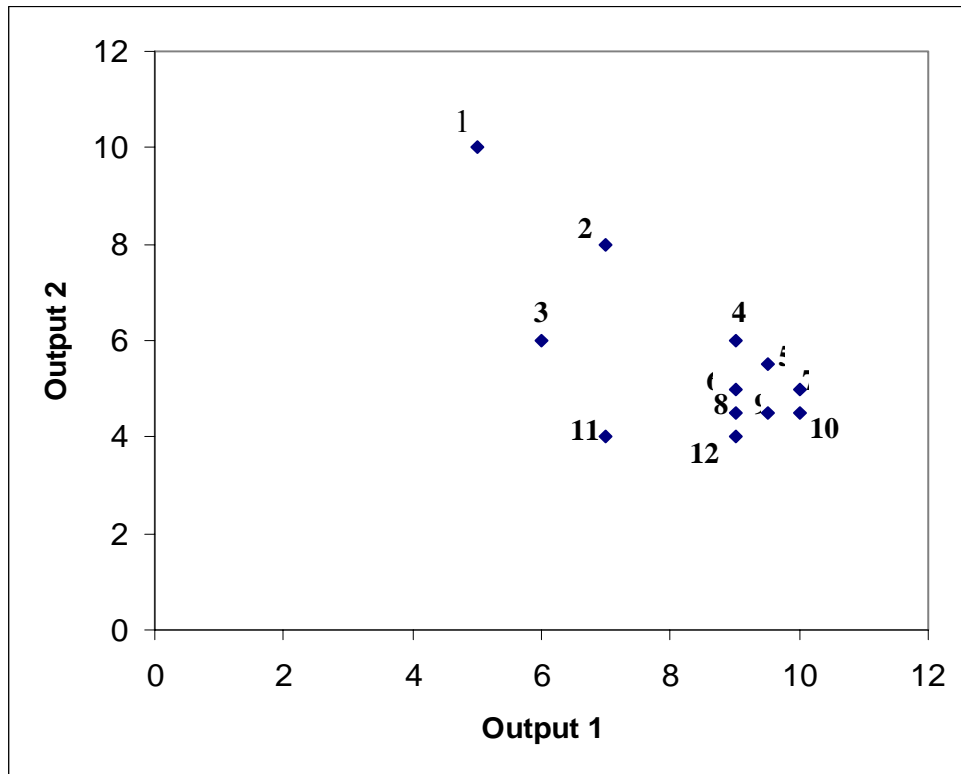


Figure 4. Graphical representation of the third example set

Table 6. The third example set in the rank order of Cross-Efficiency scores

DMU #	Cross-Efficiency Score	Output 1	Output 2
7	0.9583	10.0	5.0
5	0.9417	9.5	5.5
10	0.9375	10.0	4.5
4	0.9250	9.0	6.0
9	0.9000	9.5	4.5
6	0.8833	9.0	5.0
8	0.8625	9.0	4.5
2	0.8583	7.0	8.0
12	0.8417	9.0	4.0
1	0.7917	5.0	10.0
3	0.7000	6.0	6.0
11	0.6917	7.0	4.0

## Ranking DMUs with GTR scores

As explained in previous chapters, the Radii of Classification Preservation Method by Rousseau and Semple (1995) can be used for ranking both efficient and inefficient DMUs. For efficient DMUs, the method finds positive scores that show the minimum change required in the inputs and outputs to render the DMUs inefficient (falling short of the hypothetical frontier formed by all DMUs but the DMU in consideration). For inefficient DMUs, the scores show the minimum improvement required in inputs and outputs to make them efficient (reaching the efficient frontier). The method uses Tchebycheff distance in measuring changes, and depending on the choice of two alternative models, these changes can be computed in absolute or relative measures. We illustrate the General Tchebycheff Radii (GTR) of Classification Preservation model here, thereby finding relative changes in each DMU's inputs/outputs.

Like the use of classical DEA scores for ranking, GTR scores are also highly dependent on the efficient frontier. Assessing DMUs with GTR analysis may result in misleading results because of the sensitivity of its scores to outliers.

To illustrate our case, let us again consider the first example set of Table 2. As we have argued before, DMU 8 acts like the only obstacle for DMU 10's efficiency. DMU 10, which lies considerably inside the frontier due to DMU 8, reaches efficiency instantly once it is deleted. Table 7 shows the original GTR scores of the 12 DMUs and Table 8 shows the new scores after the removal of DMU 8. Both tables are in the rank order. The change in the rank of DMU 10 is remarkable; from the 9<sup>th</sup> position it directly jumps to be the 1<sup>st</sup>. The removal of even one DMU also affects the relative standing of the DMUs. As a method similar to the use of classical DEA scores, GTR is also highly dependent on the positioning of the DMUs relative to each other.

Another disadvantage of GTR analysis is that it is not possible to incorporate any preference information into the analysis. The weights used in this method are not on the input/output values, but on the DMUs in the set other than the DMU in consideration. This attribute prevents to include any judgments on the relative values of inputs/outputs or any preferences of the DMs on some inputs/outputs over others. So, GTR provides a ranking that is not DM specific and therefore may be of little value in most situations.

**Table 7. Initial GTR Scores for the first example set**

DMU #	GTR Score
8	0.3500
2	0.0253
1	0.0000
5	-0.0112
4	-0.0507
3	-0.0588
6	-0.1070
7	-0.2000
10	-0.3794
9	-0.6812
11	-1.0000
12	-3.0149

**Table 8. GTR Scores for the first example set without DMU 8**

DMU #	GTR Score
10	0.2418
5	0.0574
2	0.0253
1	0.0000
4	-0.0487
3	-0.0588
6	-0.1025
7	-0.2000
11	-0.3000
9	-0.5975
12	-2.5405

#### 4.2. Some Methods Developed to Rank DMUs in DEA

##### The Use of $D_k$ with a New Measure, $O_k$

As we mentioned earlier,  $D_k$  is a measure that counts the number of DMUs that should leave the set for a DMU on hand to reach efficiency.  $D_k$  distinguishes only between inefficient DMUs.

We here propose a new measure stemming from a reasoning similar to that of  $D_k$ . After the determination of the DMUs that should leave the set for each DMU in the  $D_k$  analysis, we count the number of DMUs for which each DMU had to leave the set. We name this new measure as  $O_k$ , indicating how many times a DMU on hand (efficient or inefficient) had to be deleted for the efficiency of other DMUs. For  $D_k$ , we discussed that the DMUs that should be removed for the efficiency of a certain DMU are likely to be preferred to it, and therefore units with fewer  $D_k$  values should be considered better. Likewise, the  $O_k$  value of a given DMU shows the number of DMUs that it will likely to be preferred to, and units with higher  $O_k$  values should be considered better.

Another interpretation of the  $D_k$ -  $O_k$  values is as follows: As  $D_k$  shows the number of DMUs that are preferable to a given DMU, having no additional information on hand, the best possible rank position the DMU on hand can occupy is after those DMUs. Also, as  $O_k$  shows the number of DMUs to which a DMU on

hand is preferred, the worst possible rank position the DMU on hand can occupy is before those DMUs.

The approach we here propose is: In a set of  $n$  DMUs, if  $k$  is the DMU in consideration, and  $D_k = a$  and  $O_k = b$ , then the best possible rank DMU  $k$  can acquire is  $(a+1)$  and the worst possible rank it can acquire is  $(n-b)$ . As a result, we will have a range of ranks for each DMU. Unfortunately, these ranges may be wide for most of the DMUs in the set. This will cause overlapping and intersections in the ranges which prevent ranking. Therefore,  $D_k$  and  $O_k$  values should be utilized differently for ranking purposes and two approaches will be presented on an example.

Let us use the first example set of Table 2 to illustrate our method. Table 9 shows the  $D_k - O_k$  values of the 12 DMUs, and their best and worst possible ranks.

**Table 9.  $D_k - O_k$  values of the first example set, their best and worst possible ranks**

DMU #	$D_k$	$O_k$	Best Possible Rank	Worst Possible Rank
1	0	4	1	8
2	0	6	1	6
3	2	2	3	10
4	2	4	3	8
5	1	4	2	8
6	4	3	5	9
7	6	1	7	11
8	0	6	1	6
9	7	1	8	11
10	1	3	2	9
11	2	2	3	10
12	11	0	12	12

As one can see, the ranges by themselves cannot be used as a basis for ranking. The only DMU we can position by these results is DMU 12, which will certainly be the last in the rank.

What we can do is to choose between two approaches: the optimist and the pessimist approach. The optimist approach first ranks the DMUs in ascending order of best possible rank and then breaks the ties (to the extent possible) by utilizing the worst possible ranks. The pessimist approach performs the opposite. Table 10 shows the results of the two approaches.

**Table 10. Ranking with optimist and pessimist approaches of  $D_k$  -  $O_k$**

Rank	DMU(s)	
	The Optimist Approach	The Pessimist Approach
1	2,8	2,8
2	1	1
3	5	5
4	10	4
5	4	10
6	3,11	6
7	6	3,11
8	7	7
9	9	9
10	12	12

The use of the optimist or the pessimist approach does not lead to substantial differences as one can observe.

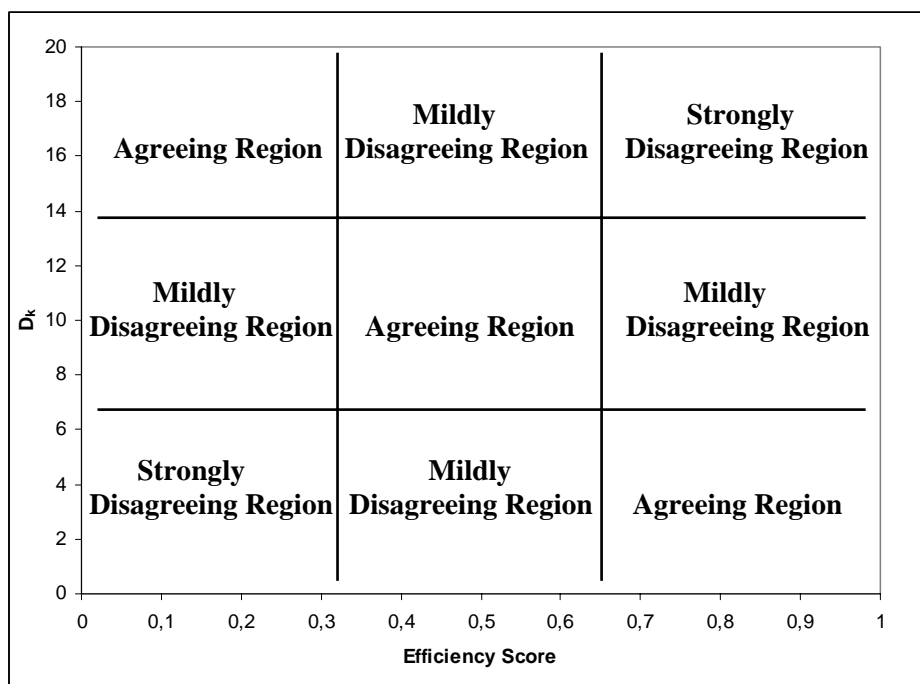
Although easy to comprehend and apply, this method has some disadvantages. First of all, some ties cannot be broken, so this method will not be suitable for cases where a complete ranking is desired. Second, it still carries the drawbacks of the use of  $D_k$  discussed in the previous section.  $O_k$  suffers from the same concerns too. If a DMU does not have to be removed for many other DMUs,  $O_k$  will judge it as a poor performer no matter what the characteristics of the DMU set are. A DMU may be a good alternative but there may just not be many DMUs in its side of the plot. But when the DMU set does not carry such properties and ties are permissible, this method may be convenient.

### **The Use of $D_k$ with DEA Efficiency Scores**

In previous discussions, we have pointed out the probable disadvantages of using DEA efficiency scores or the  $D_k$  values for ranking DMUs. Considering only the efficiency scores may lead to underrating of some DMUs because of one or few outlier DMUs. On the other hand, taking only the  $D_k$  values into consideration leads to underrating of some DMUs with respect to others which do not significantly perform better. Consider the DMUs with low efficiency scores due to a few outlier DMUs. If one looks at their  $D_k$  values, s/he would find them to be relatively low. Now consider the DMUs with high  $D_k$  values due to crowding in their area. If they

are close to the frontier, their efficiency scores will be found to be relatively high. So, what we conclude from these discussions is that, considering both the efficiency scores and the  $D_k$  values will prevent the disadvantages of considering each measure separately.

We propose here the consideration of efficiency scores and  $D_k$  values together to assess DMUs. One approach to see both measures in a common setting is the construction of a two-dimensional graph with one measure in each axis. We will use Figure 5 to discuss the implications of this graph.



**Figure 5. The general Efficiency score- $D_k$  graph**

First note that the range for the  $D_k$  values in Figure 5 are arbitrary and has no impact on the general discussion. Second, the borders for the regions are for the DM to set; they are for the separation of good, medium and poor performers in efficiency score and  $D_k$ . As one can observe, if a DMU has a high efficiency score and low  $D_k$  value, the two measures agree that this DMU is a good performer and it falls in the bottom right of the plot. Likewise, if a DMU has a low efficiency score and a high  $D_k$  value, or it has average values of both, the two measures again agree on that

DMU. All other regions are regions which contain the DMUs the measures cannot agree on, either mildly or strongly.

The DMUs in the disagreeing regions are the DMUs we want to deal with in particular. They are the units that may be suffering from the disadvantages of using only efficiency scores or  $D_k$  values which we mentioned earlier. One approach to handle assessing such DMUs is to consider only their strong side, the efficiency score or the  $D_k$  value, whichever will place them among higher positioned DMUs.

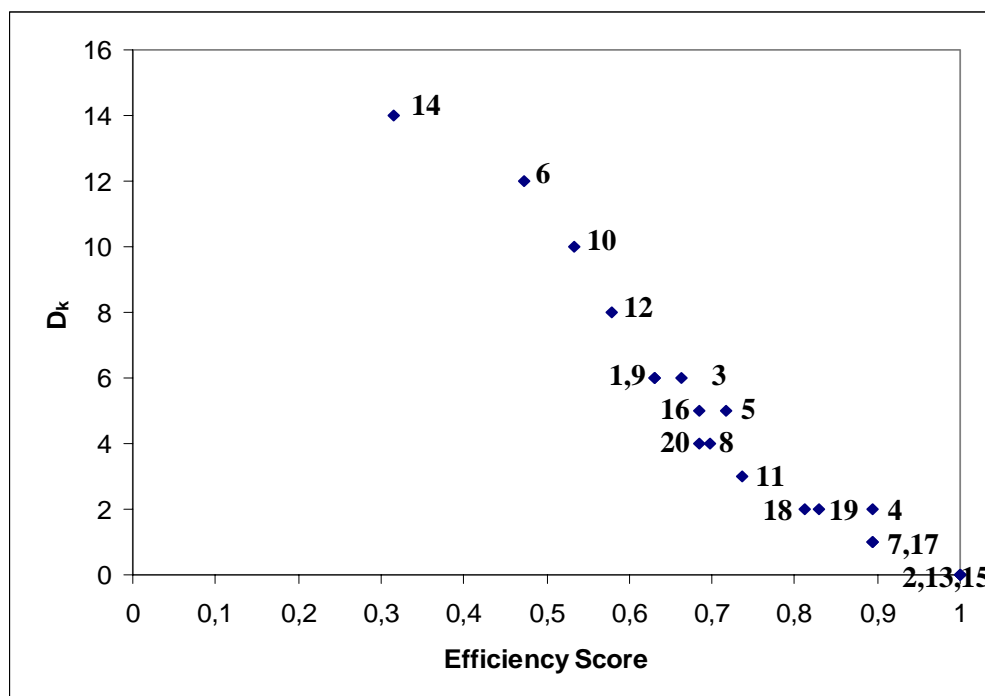
To illustrate the Efficiency score-  $D_k$  graph on an example, let us present the fourth example set. Table 11 shows the input/output values, efficiency scores and  $D_k$  values of 20 DMUs and Figure 6 shows their Efficiency score –  $D_k$  graph.

**Table 11. The fourth example set, efficiency scores and  $D_k$  values**

DMU #	Input	Output 1	Output 2	Efficiency Score	$D_k$
1	1	12	5	0.6315	6
2	1	19	11	1.0000	0
3	1	12	8	0.6637	6
4	1	17	2	0.8947	2
5	1	10	12	0.7161	5
6	1	9	2	0.4736	12
7	1	5	17	0.8947	1
8	1	13	8	0.6986	4
9	1	7	12	0.6315	6
10	1	10	6	0.5327	10
11	1	4	14	0.7368	3
12	1	6	11	0.5789	8
13	1	12	19	1.0000	0
14	1	6	3	0.3157	14
15	1	19	2	1.0000	0
16	1	3	13	0.6842	5
17	1	6	17	0.8948	1
18	1	11	14	0.8122	2
19	1	15	10	0.8296	2
20	1	13	1	0.6842	4

The performance of the DMUs increase as we diagonally move towards the bottom right of Figure 6. DMUs in this example set approximately fall in the agreeing regions, that is, the level of their performance in efficiency score mostly

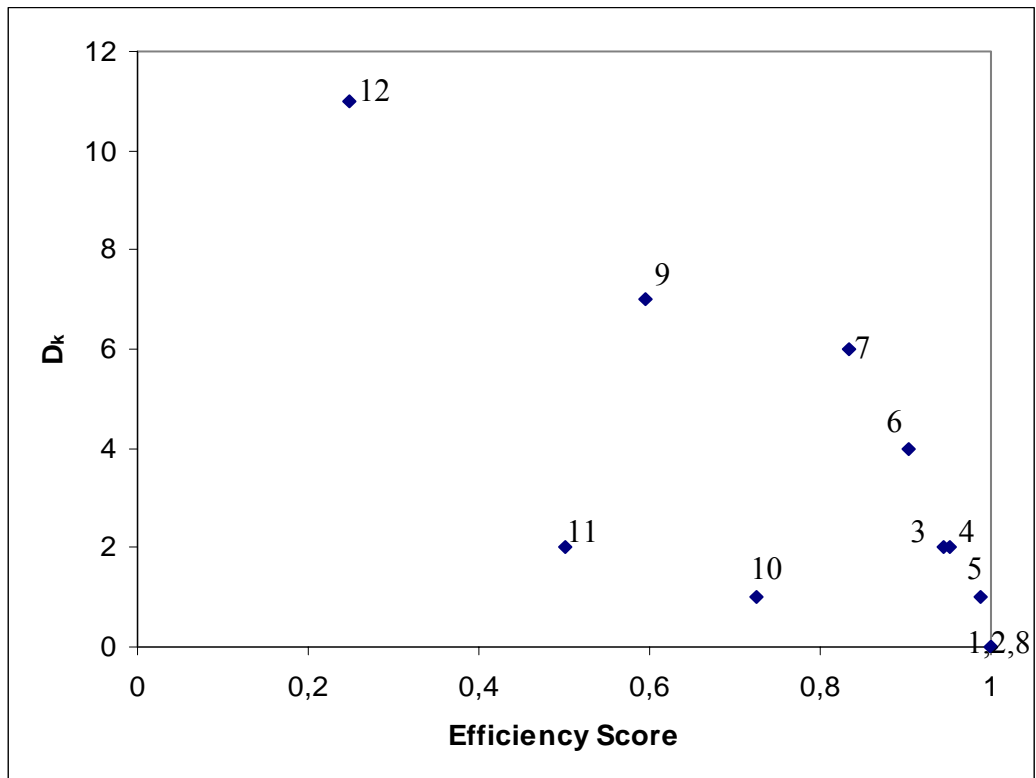
agrees with their level of performance in  $D_k$ . The reason for this is that the DMUs' output values are generated randomly in the range of 1-20, as a result, there are no outliers and there is no crowding in any area. Even in this case, DMUs are not exactly lined up along an improvement direction. This condition makes complete ranking difficult. What the DM can do is to group alternatives which s/he cannot discriminate between and then go on with ranking. Statistical clustering techniques can also be used on this graph.



**Figure 6. Efficiency Score –  $D_k$  graph of the fourth example set**

To see the deviations from the agreeing regions on an example, let us look at our familiar example set 1. Figure 7 shows the Efficiency score- $D_k$  graph. In this graph, DMUs do not line up approximately linearly as in the previous one. Especially DMUs 10 and 11, which we analyzed in previous discussions, fall apart from the central tendency. As we have proposed previously, these DMUs may be assessed by their strong sides. DMUs 10 and 11 have good levels of  $D_k$  (as opposed to their relatively low efficiency scores), so they can be regarded on the same level as the DMUs having similar  $D_k$  values (DMU 5 for 10, DMUs 3 and 4 for 11).





**Figure 7. Efficiency score –  $D_k$  graph of the first example set**

The method of using efficiency scores and  $D_k$  values simultaneously as discussed in this section is useful in the sense that it avoids the disadvantages of using a single measure that were mentioned before. However, as the examples showed, a complete ranking cannot be readily reached by this analysis as it fails to provide the DM with a final measure. This method is more suited to sorting purposes where one can apply statistical techniques.

### **Studying the change in the efficiency scores with the deletion of DMUs from the set**

As pointed out in previous discussions, the efficiency score of a DMU may not be a good indicator of its performance. DEA efficiency scores are highly dependent on the positioning of the DMUs with respect to each other. A low-scored DMU may directly reach high efficiency once one or a few number of DMUs are deleted. So, instead of looking at only the initial efficiency score of a DMU  $k$ , it will be valuable to consider its score found by deleting 0 to  $D_k$  number of DMUs iteratively. We propose here to utilize the efficiency scores of each DMU found by

deleting 0 -  $D_k$  number of DMUs from the set. The mathematical model employed for this purpose is a variant of the model ( $P_k$ ) we developed earlier. This model maximizes the efficiency score of a DMU by limiting the number of deleted DMUs by consecutive numbers between 0 and  $D_k$ , one number in each run. The model is:

$$\begin{aligned}
 & \text{Max } Z = \sum_{i=1}^s u_i y_{i0} \\
 & \text{s.t.} \\
 (P_{k-R_0}) \quad & \sum_{j=1}^m v_j x_{j0} = 1 \\
 & \sum_{i=1}^s u_i y_{ik} \leq \sum_{j=1}^m v_j x_{jk} + N_k * M \quad \text{for all } k \text{ except } 0 \\
 & \sum_{i=1}^s u_i y_{i0} \leq 1 \\
 & \sum_{k=1}^n N_k \leq R_0 \\
 & N_k \in \{0,1\} \\
 & u_i, v_j \geq 0 \quad i=1,\dots,s ; j=1,\dots,m.
 \end{aligned}$$

where  $R_0$  is the above mentioned number between 0 and  $D_0$ .

For each DMU  $k$ , one needs to solve this model for all values of  $R_k$ . Note that the optimum value for  $R_k = 0$  is equivalent to the classical DEA score and the  $R_k$  value for which an efficiency score of 1 is achieved is the  $D_k$  value. So, one does not need to know the  $D_k$  value in advance for this model, s/he just can stop when the optimum value gives 1. It is important to note that each DMU chooses the DMUs to leave the set itself; it chooses so as to make its score as high as possible.

Let us look at Table 12 which shows the changes in the efficiency scores of the 12 DMUs of the first example set with the deletion of 0 to 11 DMUs. In this table, the column number corresponding to the first '1' for each DMU in a row shows its  $D_k$  value; these 1's are seen in bold. DEA-efficient alternatives 1, 2 and 8 start with scores of 1, and 12 is the only DMU which needs all DMUs except itself to leave the set to become efficient. Remember the discussions we made on DMUs 10 and 11, here we can see how quickly they reach efficiency.

**Table 12. Efficiency scores for the first example set with the deletion of 0-11 DMUs**

		# deleted											
		0	1	2	3	4	5	6	7	8	9	10	11
DMU #	1	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	2	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	3	0.944	0.948	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	4	0.952	0.974	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	5	0.989	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	6	0.903	0.907	0.927	0.960	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	7	0.833	0.844	0.882	0.882	0.891	0.938	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000
	8	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	9	0.595	0.626	0.659	0.739	0.857	0.857	0.857	<b>1.000</b>	1.000	1.000	1.000	1.000
	10	0.725	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	11	0.500	0.769	<b>1.000</b>	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	12	0.249	0.282	0.319	0.348	0.429	0.429	0.429	0.500	0.600	0.600	0.678	<b>1.000</b>

Once we have a table like Table 12, we can logically conclude this: If DMU  $i$ 's efficiency scores are always above the ones of  $j$ , DMU  $i$  should be ranked higher than  $j$ . The reasoning is, for a DMU to have higher scores than another at every column of the table, it should necessarily start with higher scores and its  $D_k$  value should necessarily be less than or equal to that of the other. These two conditions provide the superiority of DMU  $i$  in both measures. Furthermore, its efficiency scores manage to be higher for all numbers of deleted DMUs.

Let us illustrate Table 12 on a two-dimensional graph, the Efficiency score - # of deleted DMUs graph given in Figure 8. DMUs 1, 2 and 8 start right away with scores of 1 and DMU 12 is the last to reach the 1-efficiency score line. For a DMU to have higher scores than another at every  $R_k$  value, its curve should lie above that of the other at every point in the graph, and as a result there should be no intersections. One can see the intersections in the above graph. And in this instance, they are caused by DMUs 10 and 11 we studied many times. They start with relatively low efficiency scores but make quick jumps which account for the intersections. We want to take their fast improvements into consideration.

To rank DMUs by this graph, we can start with ranking DMUs that have no intersections among themselves. In this example, the biggest such subset can be achieved by leaving DMUs 10 and 11 aside for a while. All remaining DMUs can be ranked from the best to the worst by observing their levels in the graph; the higher a DMU's line is, the higher it would be ranked. Note that, up to this point, the procedure is identical to the use of classical efficiency scores. For the intersecting

DMUs, we propose to decide on their positions by considering their  $D_k$  values and the DMUs they are intersecting. Let us illustrate our approach on this example set. Figure 9 shows first the rank order of the non-intersecting DMUs. In this rank, the left-out DMUs 10 and 11 are illustrated beside the DMUs they are intersecting. DMU 10 intersects DMUs 3, 4, 6, 7 and DMU 11 intersects DMUs 6, 7 and 9. We propose to position them to a rank among the DMUs they are intersecting. To decide on the exact position, of DMU 10 for example, we look at the  $D_k$  value of 10 and also the DMUs it intersects. We can see that the  $D_k$  of 10 is the lowest, so we place it on top of the DMUs it intersects. The situation is the same for DMU 11 too.

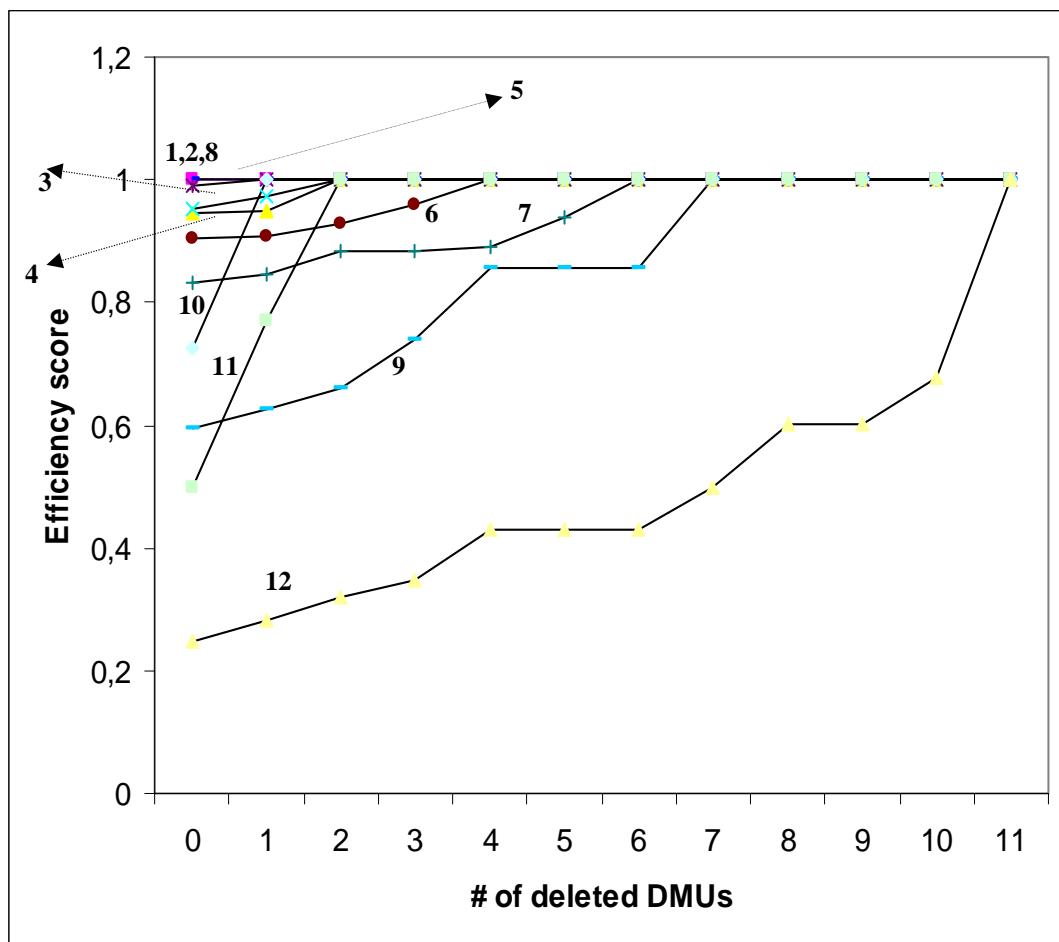
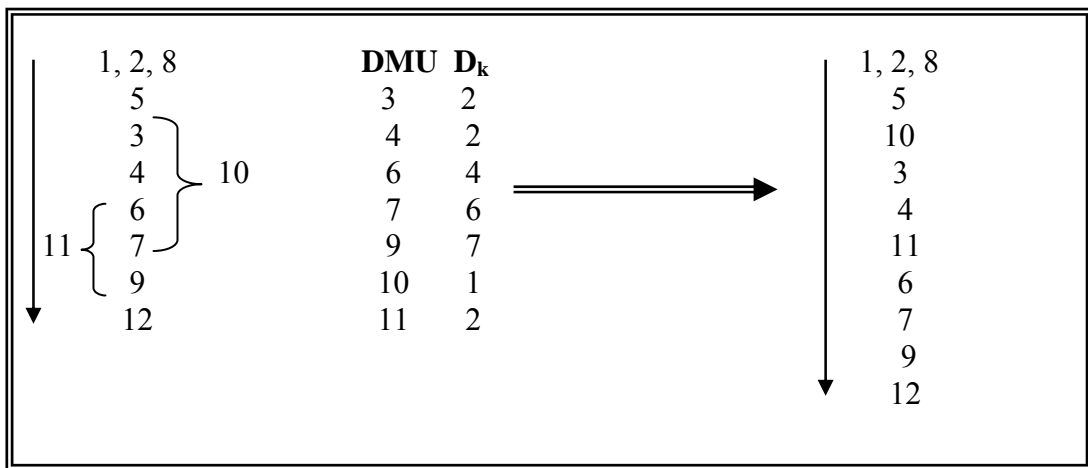


Figure 8. Efficiency score - # of deleted DMUs graph of the first example set

The basic motive behind this approach is not to rank DMUs with poor initial efficiency scores low if they manage to improve their scores quickly. The quicker a

DMU improves, the more chance it has on intersecting DMUs with high efficiency scores and low  $D_k$  values. And if two DMUs intersect on the graph, we can no longer conclude on the strict superiority of one over another and we may consider an additional measure.

This method is appropriate when the DMU set contains outliers and quick improvers. However, there may often be ties in the rank order. Especially in bigger sets, there may be many DMUs intersecting and the  $D_k$  values may not be distinct enough to allow the secondary ranking.



**Figure 9. Ranking the first example set with Efficiency score - # of deleted DMUs graph and  $D_k$  scores**

### 4.3. A New Approach to Ranking and Sorting DMUs in DEA

We have explained in the previous chapters that DEA is a non-parametric approach for measuring the efficiency scores of DMUs. It performs this task by the use of the efficient frontier formed by DMUs that are likely to outperform others. So, since it is non-parametric and it measures efficiency with respect to the efficient frontier specific to the set on hand, it is highly dependent on the characteristics of the given set of DMUs. The efficiency scores of the DMUs may be very sensitive to minor changes in the set, as illustrated in a previous section of this chapter. The use of  $D_k$ , an alternative measure to the DEA score to rank DMUs, can prevent some

misjudgments caused by the above mentioned issue, but it brings its own shortcomings.

What we will present here is a new method to rank and sort DMUs. We propose the use of efficiency scores of each DMU with the deletion of successive number of DMUs from the set, not just the initial classical score. A successful method should:

- ❖ favor DMUs whose low efficiency scores are caused by a few outlier DMUs and which manage to improve quickly once a small number of DMUs are removed from the set;
- ❖ favor DMUs whose high  $D_k$  values are caused by crowding in their area and which have good starting efficiency scores;
- ❖ always place a DMU higher in the rank than others if its scores remain above those of others as we keep removing DMUs.

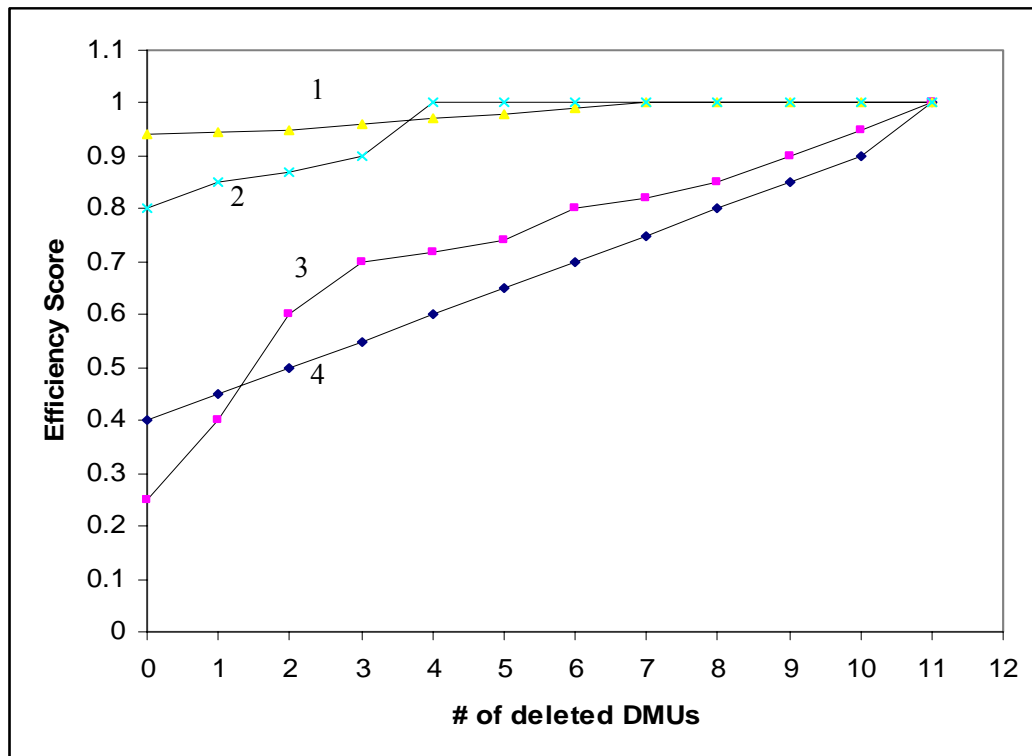
A method satisfying these conditions will eliminate the drawbacks of the aforementioned methods.

### **The Method of the Area of the Efficiency Score Graph**

Remember the “Efficiency score - # of deleted DMUs graph” in Figure 8 of the previous section. This graph illustrates how the efficiency scores of each DMU change with the deletion of  $0, 1, \dots, D_k$  DMUs sequentially; these scores are found by the  $(P_{k-R_0})$  model we developed in the previous section. The starting point for DMU  $k$  represents its original DEA efficiency score and its line climbs until it reaches the efficiency level of 1 after  $D_k$  DMUs are eliminated. This graph is sufficient to display the ‘history’ of each DMU.

What we do in our approach is to arrive at a final measure for the performance of the DMUs from this graph. Observe that efficient DMUs do not move at all in this graph; they start at 1-efficiency level and stay there. On the other hand, inefficient DMUs start at their original DEA scores and make their way up as fast as their initial scores and  $D_k$  values permit. The slope of this jump is dependent on how fast they manage to improve. A DMU that performs better than another at every  $D_k$  value necessarily lies above it. Then, a meaningful measure for the performance of the DMUs would be the area between their score curves and the 1-efficiency score line. The lower the area of a DMU, the higher it would be ranked.

All efficient DMUs will have areas of 0. We name this approach as the Area of the Efficiency Score Graph (AES). Let us look at Figure 10 to illustrate our case.

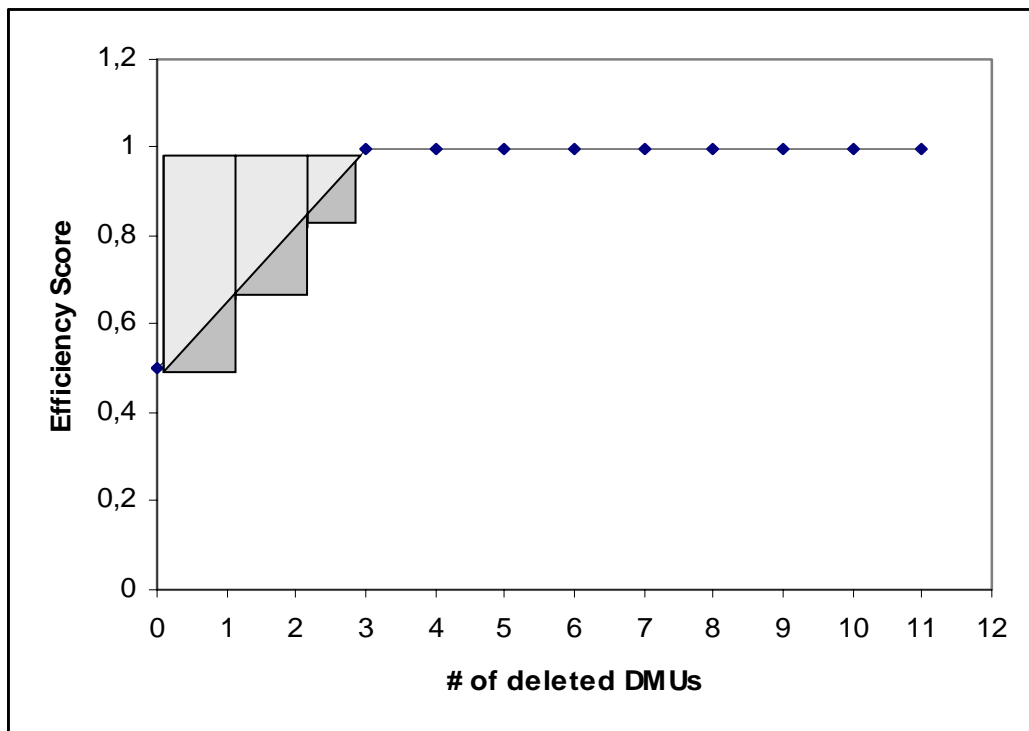


**Figure 10. Efficiency score - # of deleted DMUs graph of 4 hypothetical DMUs**

There are four hypothetical DMUs in the graph of Figure 10; they are part of a set which is not illustrated in full in the graph. Let us first study DMU 1. It starts with a high efficiency score but it cannot reach the 1-level until 7 DMUs are deleted. On the other hand, DMU 2 reaches efficiency with the deletion of 4 DMUs. The  $D_k$  approach will place DMU 2 higher than 1, but when we look at the area between these two DMUs' score curves and the 1-line, we see that DMU 1 has a smaller area because it has high efficiency scores throughout the plot. Its high  $D_k$  value is not an indicator of bad performance; it is a result of the characteristics of the DMU set. Now look at DMUs 3 and 4. We see that the original DEA score of DMU 3 is considerably lower than 4, however, it improves quickly. With the deletion of two DMUs, it manages to earn a higher efficiency score than DMU 4 and it maintains its

superiority until both reach efficiency. AES will place DMU 3 higher since it has a smaller area and therefore reward its quick improvement.

There are two ways to measure the area above the efficiency score curves of the DMUs. The first one is to measure the exact area between the score curves and the 1-line. This approach treats efficiency scores continuous; that is, it assumes that the efficiency scores continue to increase between two successive deletions. The second approach measures the areas in discrete histogram-like bars. Figure 11 illustrates the two approaches. The triangle shows the area according to the first approach, and the three bars together constitute the area according to the second approach.



**Figure 11. The areas according to two measurement approaches**

To express the areas according to the two approaches mathematically, let  $E_{kj}$  = the efficiency score of DMU  $k$  with the deletion of  $j$  DMUs,  $j = 0, \dots, D_k$  and  $A_{kl}$  be the area for DMU  $k$  computed with approach  $l$ ,  $l = 1, 2$ . Then,



$$A_{k1} = \sum_{j=0}^{Dk-1} (1 - (E_{kj} + E_{kj+1}) / 2)$$

$$A_{k2} = \sum_{j=0}^{Dk} (1 - E_{kj})$$

We prefer to use the second approach to measuring areas because efficiency scores are discrete measurements.

As an alternative measure to the area above the score curve of a DMU, area below the curve can also be used. In this case, we will be measuring efficiency (instead of inefficiency) with each deletion. Note that with this approach, higher area values will indicate better performance. If we name this approach as the third approach, area of DMU k will be (measured again in discrete steps):

$$A_{k3} = \sum_{j=0}^{n-1} E_{kj}$$

where n is the number of DMUs in the set. If one wishes to scale the scores between 0 and 1, areas can be divided by the number of DMUs. Other scales may also be used.

In our study, areas will be computed with the second approach in examples and experiments throughout the study. We will measure the areas above the score curves and not scale them.

Let us apply the AES method to some of the prior examples we used for our discussions. Remember the first example set we used first for discussing the use of classical DEA scores. DMU 8 acted as an outlier affecting the performances of DMUs 10 and 11, it affected especially DMU 10 significantly. We argued that DMUs should not be judged harshly if a few outliers account for their underperformance. Table 13 shows the ranking of the first example set by AES, it also includes the original DEA efficiency scores to make comparison possible.

As you can see, when ranked by AES, DMU 10 manages to be positioned before DMUs 6 and 7 which have higher efficiency scores. The reason for this is DMU 10's quick improvement (after the removal of DMU 8) relative to relatively slow progresses of DMUs 6 and 7. DMU 11 also is ranked higher than DMU 9 which has a higher initial efficiency score. The reason is again quick improvement.

One can refer to Table 12 and Figure 8 for the change in efficiency scores of this example set.

**Table 13. The areas and efficiency scores of the first example set**

DMU #	Area	Efficiency Score
1	0.0000	1.0000
2	0.0000	1.0000
8	0.0000	1.0000
5	0.0111	0.9888
4	0.0743	0.9516
3	0.1075	0.9444
10	0.2750	0.7249
6	0.3025	0.9033
7	0.7293	0.8333
11	0.7307	0.5000
9	1.8103	0.5947
12	6.1379	0.2490

Now let us consider the second example set. In this set, DMUs that are very close to each other in output values receive very different  $D_k$  values. The reason is their positioning with respect to each other. Despite  $D_k$ , AES will not differentiate among them significantly since they have high starting efficiency scores that are similar. And in the third example set, DMUs 1 and 7, which had very different ranks with Cross-Efficiency Method resulting from crowding in certain areas, will both have areas of 0 with AES.

The fourth example set presented in Table 11 consists of 20 randomly generated DMUs. Figure 6 shows the Efficiency Score -  $D_k$  graph we used to evaluate this set. As we have mentioned, the direction of improvement in the performances of the DMUs is towards the bottom right of the graph. Although useful in the sense that it takes the two measures into consideration, the Efficiency Score -  $D_k$  graph cannot be directly used as a ranking technique. The AES method also takes the two measures into account, and it gives a final measure that can be used for ranking and sorting. Table 14 shows the ranking of the fourth example set with the AES method; one can refer to Figure 6 to see the accord of the direction of improvement with AES ranking.

**Table 14. Ranking the fourth example set with the AES method**

<b>DMU #</b>	<b>Area</b>
2	0.0000
13	0.0000
15	0.0000
17	0.1051
7	0.1052
19	0.2059
4	0.2061
18	0.2227
11	0.6161
5	0.6670
8	0.7205
16	0.9292
20	1.0000
3	1.0760
9	1.1142
1	1.2385
12	1.7563
10	2.4909
6	3.6295
14	6.7842

The area values found by AES method are the single final measures to rank DMUs in DEA. In cases where sorting the DMUs is desired, a sorting/clustering algorithm can be performed on the area values (see Jain et al. (1999) for a review of data clustering algorithms). We propose the use of an incremental clustering algorithm. An incremental clustering algorithm adapted to our case is presented below:

#### **An Incremental Clustering Algorithm for Sorting DMUs with AES Scores**

Once we perform the AES method on the DMUs in consideration, we can rank them in the ascending order of the area values. Then we can perform the steps of the following incremental clustering algorithm:

- (1) Assign the first DMU in the rank to a class.
- (2) Consider the next DMU. Either assign it to the last class opened or assign it to a new class. Do this assignment based on the area values. If the difference between the area of the to be assigned DMU and the average area of the last class opened is less than a defined value, assign it to the existing class,

otherwise to a new class. Update the average area value of the class the DMU is assigned to.

(3) Repeat step 2 until all DMUs are placed in a class.

The first class opened will be our first ranked group and other classes will follow in order of opening.

### **The Use of Weight Restrictions with the AES Method**

The AES Method, like several other DEA-based ranking methods, provides a single ranking of DMUs when the weights are left unrestricted. Although this freedom of the weights brings the advantage of easy application, the results will not be DM-specific. A single list will be available for all stakeholders, even though they may have very different perceptions about the inputs/outputs. The DM of a problem will most probably have ideas about the ideal mix of inputs and outputs in her/his mind. S/he will also have preferences of some inputs and outputs over others. If these are not incorporated into the analysis, the resulting ranking list can be of little value. Therefore, the AES method should contain weight restrictions to enable customized ranking.

The inclusion of weight restrictions to the AES Method requires only the addition of constraints on weights to the  $(P_{k-R_0})$  model. The restrictions can be in various forms discussed before. The choice will depend on the DMU set characteristics, prior information and DM specifications. For example, when the analyzer has information on the relative values of the inputs/outputs, the use of Assurance Regions will be suitable. When the DM can express her/his preferences by percentage values, virtual weight restrictions will be more convenient.

The use of weight restrictions will also decrease the number of DMUs with areas of 0 in the AES method. When complete ranking is desired, weight restrictions will prevent multiple DMUs to already start with efficiency scores of 1.

## CHAPTER 5

### EXPERIMENTS

In this chapter, first we introduce the problem we will apply the AES method on. Then, we establish our experimental settings, report the results of the experiments and discuss the outcomes. The same problem is also solved with the Cross-Efficiency Method for comparison.

#### 5.1. The MBA-Ranking Problem

The ranking of the MBA programs offered by various schools have received growing interest. Publications like Financial Times, US News & World Report, Business Week release ranking lists of MBA programs updated yearly or every two years in their web sites. They gather information on multiple criteria they want to consider when ranking the programs. With the help of pre-assigned fixed weights on these criteria, a final score for each program is found which is used for ranking.

Financial Times (FT) ranks global MBA programs according to 20 criteria in 3 categories: Alumni career progress, diversity and idea generation. Table 15 lists these 20 criteria and their weights. The weights are in percentage values and sum up to 100 percent. FT also provides the performances of the programs in these criteria besides their final ranking. However, while for some criteria it gives the scores, for some it only provides the ranking of the programs. For example for career progress, it gives the rank positions of the programs in this criterion. Working on the 2005 final ranks and the performances of programs in individual criteria (including scores and ranks), Köksalan et al. (2006) extract consistent scores of the programs in all criteria involved. They normalize all scores (given and newly found) between 0 and 100. Table 16 shows their scores of the 81 programs for which the FT provided complete information in 2005. It also contains the FT weights of the criteria.

**Table 15. The criteria and their weights used by FT to rank MBA programs**

<b>Criteria</b>	<b>Weight%</b>
1. Weighted salary	20
2. Salary percentage increase	20
3. Value for money	3
4. Career progress	3
5. Aims achieved	3
6. Placement success	2
7. Alumni Recommendation	2
8. Employed at three months	2
<b>Alumni Career Progress Subtotal</b>	<b>55</b>
9. Women faculty	2
10. Women students	2
11. Women board	1
12. International faculty	4
13. International students	4
14. International board	2
15. International mobility	6
16. International experience	2
17. Languages	2
<b>Diversity Subtotal</b>	<b>25</b>
18. Faculty with doctorates	5
19. FT doctoral rating	5
20. FT research rating	10
<b>Idea Generation Subtotal</b>	<b>20</b>
<b>TOTAL</b>	<b>100%</b>

Table 16 provides us with the data set we will use for the application of our method. We will rank these 81 MBA programs by the AES method. In Table 16, every program has its rank number on the left of its name. For easy reference, programs will be represented by these numbers throughout the rest of the study.

Again note that the FT ranking is fixed-weights based whereas DEA models permit the DMUs to choose their own weights within the allowances of weight restrictions. So, as a DEA-based approach, AES will inevitably produce different results than the FT ranking. However, it will still take the weights provided into account since they represent judgments on the relative importance of the criteria. The ways to utilize these fixed weights in AES will be discussed in the next section.

An important characteristic of this MBA set is that all criteria are in the form of outputs. The reason is that the stakeholders of this ranking are not interested in how much the programs consume, their primary concern is the outputs. So, to fit this set to DEA modeling, it can reasonably be considered as a single input (of 1) case.

**Table 16. The 81 MBA Programs with their FT ranks and scores in 20 criteria**

FT Rank	Program	Criteria and their weights																			
		Weighted salary 20%	Salary percentage increase 20%	Value for money 3%	Career progress 3%	Aims achieved 3%	Placement success 2%	Employment at three months 2%	Alumni recommend 2%	Women faculty 2%	Women students 2%	Women board 1%	International faculty 4%	International students 4%	International board 2%	International mobility 6%	International experience 2%	Languages 2%	Faculty with doctorates 5%	FT doctoral rating 5%	FT research 10%
1	University of Pennsylvania: Wharton	88.35	77.78	25.23	47.00	64.71	80.65	73.17	100.00	33.33	65.71	14.04	35.37	37.78	52.00	23.05	32.88	0.00	100.00	95.06	85.65
2	Harvard Business School	100.00	65.66	24.83	96.07	47.06	54.41	78.05	99.60	46.67	68.57	29.82	34.15	26.67	14.00	32.47	0.00	0.00	91.38	84.25	100.00
3	Columbia Business School	77.12	98.99	26.85	95.26	58.82	84.29	80.49	97.17	20.00	65.71	15.79	60.98	30.00	36.00	30.85	7.68	0.00	91.38	51.63	71.76
4	Stanford University GSB	99.41	60.61	2.42	82.92	82.35	79.84	80.49	99.19	26.67	71.43	28.07	39.02	23.33	20.00	24.27	26.82	0.00	98.28	59.25	84.84
5	London Business School	73.31	72.73	3.23	80.90	70.59	35.10	65.85	97.98	16.67	37.14	14.04	87.80	86.67	65.00	64.28	57.75	50.00	96.55	33.75	54.60
6	University of Chicago GSB	82.53	83.84	6.47	43.36	52.94	98.79	73.17	97.58	16.67	45.71	28.07	40.24	26.67	11.00	21.44	34.49	0.00	96.55	43.87	85.25
7	Insead	74.95	25.25	89.73	96.88	52.94	33.88	65.85	98.38	23.33	25.71	15.79	100.00	87.78	73.00	64.68	84.65	100.00	93.10	32.54	70.55
8	New York University: Stern	65.98	81.82	2.83	32.61	58.82	82.67	78.05	96.36	33.33	65.71	22.81	41.46	22.22	3.00	21.84	30.86	0.00	100.00	86.68	67.07
9	Yale School of Management	74.96	100.00	26.44	43.76	76.47	82.27	56.10	64.67	16.67	68.57	35.09	31.71	22.22	11.00	32.06	6.87	0.00	96.55	2.42	54.20
10	Northwestern University: Kellogg	77.72	61.62	0.00	79.28	70.59	98.38	80.49	98.79	40.00	54.29	14.04	21.95	23.33	13.00	22.24	34.90	0.00	93.10	59.65	70.95
11	Iese Business School	39.52	81.82	32.91	95.66	64.71	41.56	92.68	67.09	16.67	34.29	17.54	34.15	72.22	81.00	65.09	83.84	50.00	98.28	26.46	35.72
12	MIT: Sloan	76.35	49.49	4.45	40.13	47.06	79.44	78.05	96.77	33.33	60.00	22.81	14.63	27.78	24.00	19.64	28.84	0.00	100.00	85.06	79.09
13	UC Berkeley: Haas	68.66	46.46	26.04	94.85	64.71	77.82	82.93	94.34	43.33	48.57	31.58	34.15	36.67	8.00	34.08	28.43	0.00	100.00	52.04	71.36
14	University of Michigan: Ross	67.10	62.63	4.85	41.34	76.47	97.98	70.73	95.96	53.33	60.00	64.91	32.93	30.00	10.00	15.49	26.41	0.00	93.10	50.82	65.82
15	University of North Carolina: Kenan-Flagler	64.41	79.80	31.29	30.19	82.35	83.07	56.10	67.50	33.33	54.29	19.30	25.61	22.22	4.00	19.23	26.01	0.00	93.10	27.27	64.62
16	Duke University: Fuqua	65.93	64.65	1.62	42.96	70.59	99.60	87.80	95.55	33.33	54.29	22.81	39.02	25.56	6.00	24.67	32.47	0.00	96.55	32.13	65.02
17	University of Virginia: Darden	73.87	81.82	27.65	93.24	76.47	99.19	56.10	95.15	46.67	31.43	35.09	10.98	18.89	8.00	23.46	4.45	0.00	94.83	14.32	20.79
18	University of Toronto: Rotman	42.17	70.71	90.14	46.19	76.47	37.92	78.05	67.90	40.00	51.43	66.67	65.85	35.56	46.00	25.07	9.70	0.00	94.83	29.70	59.32
19	York University: Schulich	30.89	75.76	94.16	97.68	47.06	7.27	73.17	63.05	56.67	71.43	38.60	62.20	68.89	49.00	46.35	36.52	0.00	98.28	2.83	44.55
20	Cornell University: Johnson	67.63	56.57	0.81	77.26	64.71	83.88	60.98	63.86	60.00	51.43	26.32	31.71	32.22	46.00	17.62	27.62	0.00	93.10	2.42	64.22
21	University of Oxford: Said	59.75	44.44	100.00	99.60	58.82	28.71	60.98	49.43	26.67	11.43	26.32	39.02	87.78	26.00	60.51	31.26	0.00	96.55	1.21	45.36
22	UCLA: Anderson	68.37	47.47	6.87	30.99	41.18	80.25	73.17	94.75	13.33	62.86	15.79	23.17	21.11	9.00	18.02	33.28	0.00	100.00	29.29	84.44
23	University of Rochester: Simon	48.21	82.83	6.06	42.55	70.59	79.03	75.61	13.11	16.67	57.14	26.32	32.93	43.33	25.00	33.28	5.66	0.00	87.93	16.75	46.17
24	Rotterdam School of Management	46.11	48.48	69.02	79.69	47.06	38.33	68.29	61.84	23.33	48.57	47.37	19.51	97.78	40.00	63.88	99.60	0.00	82.76	50.02	19.57
25	University of Maryland: Smith	45.12	72.73	30.48	30.59	64.71	29.92	75.61	49.03	53.33	68.57	17.54	23.17	34.44	20.00	16.30	1.21	0.00	100.00	29.70	63.41
26	Vanderbilt University: Owen	51.42	82.83	28.46	94.45	58.82	40.35	68.29	54.61	40.00	51.43	14.04	14.63	25.56	8.00	45.54	5.66	0.00	96.55	1.62	21.19
27	Carnegie Mellon: Tepper	53.53	59.60	5.25	13.57	64.71	83.48	80.49	65.07	23.33	28.57	14.04	35.37	21.11	8.00	16.71	3.64	0.00	96.55	86.27	52.45
28	University of Western Ontario: Ivey	44.44	58.59	47.10	39.72	47.06	34.29	43.90	66.69	36.67	42.86	22.81	35.37	54.44	44.00	45.13	56.94	0.00	93.10	14.32	53.39
29	University of Iowa: Tippie	36.44	80.81	49.53	78.07	70.59	46.47	80.49	5.14	36.67	62.86	29.82	13.41	41.11	0.00	32.87	6.06	0.00	91.38	26.06	34.91
30	Esade Business School	30.85	79.80	30.89	84.13	64.71	36.71	73.17	15.94	43.33	28.57	57.89	15.85	68.89	100.00	65.49	85.05	100.00	51.72	2.83	1.62
31	University of Southern California: Marshall	49.27	65.66	4.04	39.32	64.71	81.86	39.02	62.25	53.33	42.86	26.32	24.39	18.89	27.00	18.83	30.05	0.00	67.24	26.06	63.81
32	HEC Paris	42.01	48.48	70.63	98.49	64.71	81.05	0.00	49.43	26.67	34.29	26.32	21.95	73.33	75.00	60.91	58.16	100.00	53.45	27.27	17.55
33	McGill University	32.38	46.46	79.95	83.33	52.94	38.73	19.51	63.46	66.67	60.00	33.33	71.95	45.56	28.00	61.32	35.30	0.00	81.03	3.64	52.86
34	Lancaster University Management School	26.37	53.54	87.31	99.19	52.94	35.50	75.61	6.75	56.67	42.86	100.00	25.61	84.44	43.00	45.94	58.56	0.00	74.14	84.66	10.79
35	University of Cambridge: Judge	44.79	19.19	71.04	45.38	35.29	37.12	63.41	15.53	50.00	37.14	52.63	60.98	92.22	46.00	59.70	33.69	0.00	91.38	85.06	42.53
36	SDA Bocconi	22.70	63.64	71.85	44.98	35.29	30.33	53.66	15.13	96.67	42.86	71.93	14.63	37.78	24.00	56.65	60.18	100.00	67.24	85.87	21.59
37	Manchester Business School	43.20	55.56	46.30	44.17	100.00	36.31	56.10	62.65	26.67	42.86	0.00	25.61	82.22	0.00	58.08	84.25	0.00	72.41	32.13	10.39
38	Boston University School of Management	45.80	72.73	2.02	21.06	58.82	6.47	75.61	14.72	56.67	91.43	26.32	26.83	33.33	15.00	23.86	28.03	0.00	79.31	26.87	43.74
39	Hong Kong UST Business School	6.74	36.36	71.44	97.28	52.94	46.87	80.49	8.77	20.00	97.14	68.42	82.93	46.67	94.00	47.56	100.00	50.00	100.00	15.13	60.12
40	University of Illinois at Urbana-Champaign	28.09	65.66	25.63	32.21	29.41	41.97	90.24	12.30	53.33	42.86	26.32	26.83	44.44	0.00	25.07	2.83	0.00	94.83	85.46	50.43
41	Michigan State University: Broad	33.49	70.71	46.70	0.81	70.59	100.00	56.10	12.70	46.67	62.86	22.81	0.00	30.00	4.00	6.16	8.49	0.00	89.66	43.47	58.51

Table 16 (continued)

		Criteria and their weights																			
FT Rank	Program	Weighted salary 20%	Salary percentage increase 20%	Value for money 3%	Career progress 3%	Aims achieved 3%	Placement success 2%	Employment at three months 2%	Alumni recommend 2%	Women faculty 2%	Women students 2%	Women board 1%	International faculty 4%	International students 4%	International board 2%	International mobility 6%	International experience 2%	Languages 2%	Faculty with doctorates 5%	FT doctoral rating 5%	FT research 10%
42	Case Western Reserve: Weatherhead	32.59	51.52	32.10	80.09	58.82	34.69	70.73	6.35	46.67	74.29	21.05	40.24	24.44	4.00	44.73	27.22	0.00	100.00	32.54	51.64
43	University of Minnesota: Carlson	32.35	70.71	30.08	12.77	58.82	40.75	82.93	12.70	50.00	28.57	26.32	13.41	28.89	2.00	45.54	24.39	0.00	87.93	28.08	51.24
44	Warwick Business School	42.56	26.26	86.50	41.74	29.41	2.02	78.05	64.27	90.00	14.29	36.84	37.80	77.78	32.00	35.30	57.35	50.00	79.31	94.66	16.34
45	Imperial College London: Tanaka	46.21	26.26	88.92	83.73	41.18	27.90	2.44	13.91	63.33	68.57	63.16	71.95	66.67	45.00	31.66	2.02	0.00	91.38	27.68	42.93
46	Pennsylvania State: Smeal	34.38	80.81	47.51	1.62	52.94	54.82	82.93	13.51	33.33	34.29	29.82	10.98	30.00	2.00	17.21	4.85	0.00	74.14	15.94	52.05
47	University of British Columbia: Sauder	24.40	42.42	89.33	82.52	11.76	6.87	82.93	0.00	30.00	77.14	36.84	74.39	67.78	5.00	60.10	36.11	0.00	82.76	28.89	44.15
48	University of Texas at Austin: McCombs	53.96	42.42	7.27	31.80	52.94	78.22	60.98	65.88	63.33	28.57	24.56	14.63	20.00	3.00	8.59	25.20	0.00	82.76	51.23	55.00
49	Cranfield School of Management	60.50	29.29	88.52	84.54	70.59	39.94	65.85	65.48	66.67	17.14	73.68	9.76	65.56	25.00	35.70	7.27	0.00	37.93	50.42	11.60
50	Virginia Tech: Pamplin	20.69	91.92	48.72	98.09	58.82	41.16	31.71	0.00	60.00	45.71	14.04	3.66	31.11	7.00	7.37	0.00	0.00	89.66	32.54	19.98
51	City University: Cass	41.48	33.33	87.71	80.50	58.82	5.25	70.73	11.89	46.67	31.43	29.82	41.46	70.00	29.00	34.89	31.67	0.00	84.48	44.28	16.74
52	University of Arizona: Eller	25.42	69.70	49.12	0.40	88.24	39.54	68.29	5.54	73.33	65.71	24.56	15.85	37.78	1.00	0.40	0.00	0.00	87.93	52.44	36.12
53	Melbourne Business School	51.61	31.31	47.91	42.15	0.00	0.00	19.51	61.44	50.00	42.86	35.09	56.10	81.11	13.00	58.49	56.54	0.00	100.00	0.00	22.45
54	University of California at Irvine	35.99	46.46	29.27	29.78	58.82	35.90	63.41	11.49	83.33	62.86	19.30	32.93	31.11	15.00	5.76	25.60	0.00	93.10	14.32	58.91
55	Washington University: Olin	42.26	49.49	0.40	44.57	52.94	77.42	53.66	13.51	30.00	31.43	21.05	47.56	28.89	0.00	15.90	3.23	0.00	94.83	2.42	53.79
56	Arizona State University: Carey	32.47	55.56	31.70	33.02	58.82	42.37	80.49	14.32	43.33	25.71	0.00	13.41	20.00	0.00	33.68	5.25	0.00	86.21	44.68	44.95
57	Chinese University of Hong Kong	0.00	82.83	50.34	2.02	5.88	6.06	60.98	9.18	33.33	65.71	35.09	58.54	75.56	55.00	7.78	60.98	0.00	87.93	3.23	46.57
58	University of Pittsburgh: Katz	30.71	64.65	33.31	12.36	17.65	37.52	53.66	9.18	30.00	54.29	8.77	17.07	45.56	23.00	31.25	0.40	0.00	93.10	26.87	35.31
59	Tulane University: Freeman	36.54	62.63	1.21	31.40	58.82	29.12	73.17	5.94	40.00	54.29	12.28	20.73	25.56	3.00	30.45	8.89	0.00	75.86	14.32	28.60
60	Temple University: Fox	37.07	30.30	86.90	96.47	17.65	2.42	53.66	6.35	40.00	31.43	40.35	10.98	26.67	46.00	59.30	59.37	0.00	86.21	32.94	19.17
61	Queen's School of Business	35.84	39.39	81.16	13.98	70.59	78.63	43.90	66.29	60.00	28.57	19.30	45.12	52.22	26.00	18.42	0.00	0.00	81.03	2.02	17.96
62	Bradford School of Management: Nimbas	25.04	30.30	99.60	40.94	29.41	5.66	87.80	6.35	100.00	54.29	35.09	26.83	85.56	30.00	56.25	30.45	0.00	43.10	43.47	9.98
63	University of South Carolina: Moore	21.06	58.59	28.06	20.65	5.88	4.45	34.15	7.56	23.33	60.00	19.30	8.54	18.89	3.00	46.75	60.58	0.00	82.76	32.13	43.34
64	Ohio State University: Fisher	26.92	42.42	29.68	14.69	17.65	27.50	80.49	13.91	43.33	40.00	21.05	12.20	31.11	2.00	6.97	29.65	0.00	84.48	43.47	59.72
65	Concordia University: Molson	20.80	42.42	70.23	0.00	76.47	1.21	24.39	9.58	43.33	37.14	52.63	81.71	46.67	25.00	32.87	35.71	0.00	91.38	15.53	12.41
66	University of Georgia: Terry	26.75	49.49	48.32	2.42	47.06	29.52	75.61	9.58	50.00	48.57	0.00	7.32	25.56	0.00	7.78	0.81	0.00	94.83	50.02	29.01
67	University of Cape Town	52.85	34.34	93.76	77.67	35.29	1.62	41.46	13.51	50.00	51.43	12.28	20.73	18.89	36.00	55.84	83.44	0.00	0.00	0.40	1.21
68	Australian Graduate School of Management	40.81	3.03	27.25	13.17	5.88	28.31	60.98	54.20	46.67	42.86	15.79	56.10	48.89	15.00	55.44	59.77	0.00	93.10	15.13	45.76
69	Georgia Institute of Technology	29.99	43.43	32.50	81.31	47.06	55.22	82.93	11.49	23.33	17.14	28.07	12.20	36.67	0.00	20.04	8.08	0.00	81.03	14.72	22.85
70	University College Dublin: Smurfit	31.82	11.11	94.57	78.48	17.65	3.64	100.00	6.75	50.00	48.57	22.81	31.71	41.11	58.00	58.89	9.29	50.00	82.76	14.72	12.00
71	Coppead	20.10	51.52	69.83	11.96	70.59	81.46	58.54	5.54	90.00	97.14	50.88	8.54	0.00	0.00	0.00	56.13	50.00	100.00	1.21	0.00
72	Texas A & M University: Mays	24.18	42.42	80.35	1.21	52.94	47.28	92.68	12.30	60.00	51.43	7.02	1.22	16.67	2.00	8.18	1.62	0.00	77.59	33.34	36.53
73	Tec de Monterrey - Egade	17.84	37.37	28.87	93.64	23.53	4.85	70.73	4.73	56.67	37.14	17.54	13.41	64.44	0.00	22.65	58.96	50.00	94.83	0.81	9.58
74	Solvay Business School	6.37	28.28	88.12	81.71	58.82	46.87	14.63	5.14	0.00	65.71	43.86	43.90	62.22	0.00	97.30	24.80	0.00	70.69	0.81	0.40
75	University of Durham Business School	25.34	8.08	79.54	100.00	47.06	0.81	80.49	7.16	30.00	51.43	22.81	51.22	84.44	20.00	6.57	6.47	50.00	70.69	15.94	18.76
76	Edinburgh University Management School	20.46	4.04	50.74	94.05	58.82	3.23	60.98	8.37	40.00	48.57	31.58	37.80	72.22	27.00	47.15	32.07	50.00	68.97	16.34	20.38
77	Birmingham Business School	15.57	12.12	49.93	45.78	41.18	78.63	17.07	4.73	60.00	85.71	87.72	18.29	100.00	50.00	55.44	23.99	0.00	67.24	1.62	18.36
78	University of Washington Business School	24.90	18.18	5.66	40.53	35.29	39.14	97.56	0.00	76.67	42.86	15.79	3.66	18.89	4.00	15.09	34.09	0.00	81.03	44.28	50.84
79	University of Bath School of Management	25.76	1.01	69.42	46.59	17.65	2.83	100.00	11.49	63.33	100.00	29.82	29.27	67.78	22.00	34.49	29.24	0.00	91.38	27.68	17.15
80	George Washington University	27.65	22.22	3.64	78.88	47.06	4.04	48.78	7.97	60.00	74.29	12.28	6.10	31.11	5.00	45.54	2.42	0.00	93.10	28.48	11.20
81	Trinity College Dublin	33.70	0.00	80.75	82.11	70.59	0.40	2.44	6.35	96.67	0.00	57.89	29.27	33.33	0.00	65.90	4.04	0.00	58.62	0.00	0.81



## **5.2. Weight Restrictions in Ranking MBA Programs by the AES Method**

As we have mentioned in the previous section, the criteria in MBA ranking problem are all in the form of outputs. Because inputs are not of interest to the stakeholders and all programs should be given an equal starting point in terms of inputs, all programs are assigned a single input of 1. Since weighted inputs will be equated to 1, the weight for the single input is 1 for all programs and hence there exists no restriction on it.

There are 20 outputs and they are given fixed weights which sum up to 100% in the FT ranking. There are two types of weight restrictions that can be employed to incorporate these judgments into the AES Method; both are discussed in Chapter 3. The first one is the use of virtual weight restrictions and the second is the use of Assurance Regions. Virtual weight restrictions are suited to this problem in the sense that they allow the DM to specify her/his judgments in the form of percentages. We have the exact percentages which we can use to generate ranges for the importance of individual criteria. Assurance Regions can also be used in this problem, they are for relating the importance of criteria. The percentage values can reasonably be converted to relative values of the weights and again ranges can be formed. First the application of virtual weight restrictions to the MBA ranking problem and then the Assurance Regions will be studied in detail next.

### **Virtual Weight Restrictions in the AES Method for the MBA Ranking Problem**

Virtual weight restrictions impose constraints on weighted inputs/outputs rather than directly on inputs/outputs. The ratio of a weighted input (output) to the sum of all weighted inputs (outputs) is constrained by lower and upper bounds in the form of percentages. The different ways to incorporate these constraints are discussed in Chapter 3. Virtual constraints can be applied to only the DMU on hand, all DMUs in every DMU's formulation, or to the DMU on hand and the average DMU. Sarrico and Dyson (2004) point out the potential infeasibility problems with the virtual weight restrictions.

The MBA ranking problem contains fixed weights (in terms of percentages) assigned to each criterion. To apply virtual weight restrictions to the AES Method, we will generate ranges from these fixed percentages. Then there is the decision of

where to apply the restrictions. Applying the constraints to all programs in every formulation imposes very strict restrictions and will result in high  $D_k$  values close to each other for the majority of programs. Moreover, as pointed out by Sarrico and Dyson, the restrictions imposed on the DMUs will be the result of another DMU's input/output values. The other approach, imposing constraints on the DMU on hand and the average DMU results in restricting the DMUs by the 'average' behavior which does not fit to DEA basics. Therefore, we will apply the restrictions to the DMU on hand in each formulation. Remember that in our data set, all input/output values are normalized between 0 and 100. So, in a DMU's formulation, its set of weights to satisfy the range will not be inconvenient for the other DMUs to keep their weighted outputs below 1.

The model to be solved for the AES Method with virtual weight restrictions then will be:

$$\begin{aligned}
 & \text{Max } Z = \sum_{i=1}^s u_i y_{i0} \\
 & \text{s.t.} \\
 (\text{V-P}_{k-R_0}) \quad & \sum_{j=1}^m v_j x_{j0} = 1 \\
 & \sum_{i=1}^s u_i y_{ik} \leq \sum_{j=1}^m v_j x_{jk} + N_k \times M \quad \text{for all } k \text{ except } 0 \\
 & \sum_{i=1}^s u_i y_{i0} \leq 1 \\
 & a_p \leq u_p y_{p0} / \sum_{i=1}^s u_i y_{i0} \leq b_p \quad \text{for } p = 1, \dots, s \\
 & \sum_{k=1}^n N_k \leq R_0 \\
 & N_k \in \{0, 1\} \\
 & u_i, v_j \geq 0 \quad i=1, \dots, s ; j=1, \dots, m.
 \end{aligned}$$

where  $R_0$  is a number between 0 and  $D_0$  and  $M$  is a sufficiently large constant that allows the weighted outputs of DMUs to become as high as they desire. This model will be solved for every DMU  $k$  for all values of  $R_k$ .

To see the results of this (V-P<sub>k-R0</sub>) model on a smaller set, let us first apply it to the first 20 programs of the FT list. The weights are converted to ranges by increasing and decreasing them by 25%. 40% changes will also be considered to see the effects of wider ranges. Table 17 shows the FT weights and the respective ranges generated for the outputs by  $\mp 25\%$  and  $\mp 40\%$  changes. To apply virtual weight restrictions to the MBA set, there is an adjustment we need to make to the data. As it can be observed from Table 16, some programs have 0 values for some of the outputs. With these 0's, it is impossible for them to satisfy the lower bounds, no matter how large they assign the respective weights. So, we assign a very small number (0.01) to the places of the 0's.

Let us first look at the results with the  $\mp 25\%$  ranges. Table 18 shows the first 20 programs in the rank order of their area values of the AES Method; it also reveals their  $D_k$  values and efficiency scores before any programs are removed from the set. The only program that has a classical efficiency score of 1 with these weight restrictions is Program 3. Its area value is 0, and others' areas increase in the order of their ranks. The most interesting result in this ranking list is how low programs 7, 8 and especially 9 are placed. As mentioned, the AES method will inevitably produce different results than the FT list; however, when we look at the two ranking lists, we can see that they are quite close except for the mentioned programs. Let us study Program 9's output values to search for the reason. We see that its performance is very poor in Criterion 19 compared to other programs (its score is 2.4 whereas the average for 20 programs is 43.6) except for programs 19 and 20 which are already in the bottom of the FT list. If a program performs very poorly in an output, it has to assign a very large weight to that output to reach the lower bound, which in turn causes other programs' weighted outputs to exceed 1. In the AES formulations, all such programs eventually leave the set making the  $D_k$  of the program on hand very high. And since the number of programs to leave is limited by  $R_k$  in each formulation, it cannot keep its efficiency high while keeping the weighted outputs of the other programs (which are not allowed to leave by the current  $R_k$ ) below 1. To see if the explained situation is the reason for Program 9's high area value, let us artificially raise its 19<sup>th</sup> output score from 2.4 to 85. Its  $D_k$  value drops to 6 from 17 and its initial efficiency score becomes 0.890 (as opposed to 0.395). It is clear that it will receive a much lower area with an improved Output 19.

**Table 17. Ranges for the virtual weight restrictions –  $\mp$  25% and  $\mp$  40%**

Output	FT weight	AES Range – $\mp$ 25%	AES Range - $\mp$ 40%
1	20	15-25%	12-28%
2	20	15-25%	12-28%
3	3	2.25-3.75%	1.8-4.2%
4	3	2.25-3.75%	1.8-4.2%
5	3	2.25-3.75%	1.8-4.2%
6	2	1.5-2.5%	1.2-2.8%
7	2	1.5-2.5%	1.2-2.8%
8	2	1.5-2.5%	1.2-2.8%
9	2	1.5-2.5%	1.2-2.8%
10	2	1.5-2.5%	1.2-2.8%
11	1	0.75-1.25%	0.6-1.4%
12	4	3-5%	2.4-5.6%
13	4	3-5%	2.4-5.6%
14	2	1.5-2.5%	1.2-2.8%
15	6	4.5-7.5%	3.6-8.4%
16	2	1.5-2.5%	1.2-2.8%
17	2	1.5-2.5%	1.2-2.8%
18	5	3.75-6.25%	3-7%
19	5	3.75-6.25%	3-7%
20	10	7.5-12.5%	6-14%

**Table 18. AES ranking of the first 20 programs with virtual weight restrictions ( $\mp$  25%)**

Rank	Program	Area - Virtual ( $\mp$ 25%)	Dk	Efficiency scores with no deletion
1	3	0.0000	0	1.0000
2	1	0.0046	1	0.9954
3	2	0.0265	3	0.9819
4	5	0.0973	3	0.9580
5	4	0.2287	4	0.9345
6	6	0.6037	7	0.8717
7	10	0.9819	9	0.8089
8	13	1.2808	12	0.7957
9	11	1.4443	13	0.7760
10	18	1.6487	14	0.7810
11	14	1.6549	14	0.7653
12	12	1.7795	16	0.7511
13	16	1.9501	16	0.7422
14	8	1.9585	14	0.6943
15	7	2.0137	17	0.7493
16	15	2.8152	17	0.6783
17	17	4.6636	19	0.6113
18	9	6.6228	17	0.3950
19	20	7.2869	19	0.3806
20	19	7.8173	19	0.3851

As Program 9's situation demonstrated, if a program has a significantly low score even in one output, the lower bounds of the virtual weight restrictions judge it very harshly. This output may constitute a small percentage (like the 5% of Output 19). Also, the lower bounds do not allow the programs to have 0 values in any criterion. Therefore, we propose to use virtual weight restrictions with only upper bounds. This will enable the programs to stress their strong sides more freely. Unless the ranges are very wide, the use of upper bounds will result in implicit lower bounds too. For its strong sides, after a program increases the weighted outputs to their limit, it will still have to assign weights to its weak sides to fill up the 100%. For the reasons discussed, virtual weight restrictions will be applied with only upper bounds in all experiments for the rest of the study.

Let us rank the 20 programs again, this time with no lower bounds in virtual weight restrictions. Since the lower bounds are removed, there is no need to replace the 0 values in the outputs by small numbers. The output values of all programs are left unchanged throughout the rest of this study. Table 19 shows the new ranking list of the 20 programs with virtual weight restrictions (+25%) after the lower bounds are removed. The  $D_k$  values and the original efficiency score are again included for comparison. One can see the improvements in the ranks of programs 7, 8 and 9.

To observe the effect of different ranges, let us now rank the 20 programs with the upper bounds made by 40% increase in the FT weights. Table 20 shows the ranking list of the AES Method with +40% upper bounded virtual weight restrictions. The programs with an area of 0 increased from 4 to 5 when the upper bounds are formed by 40% increase. This result is expected since +40% upper bounds are less strict than +25%. Again stemming from the same reason, the initial efficiency scores of all programs are higher and their  $D_k$  values are lower. An interesting result of the +40% ranges is the improvement Program 19 makes. From the 16<sup>th</sup> position, it jumps to be the 5<sup>th</sup>. When we study the output values of Program 19, we see that it has scores high above the average in outputs 3, 4, 9, 10, 12 and 13; in fact, it has the highest scores (among 20 programs) in outputs 3, 4, 9 and 10. By increasing the upper bounds, we give Program 19 more space to stress its strengths.

The decision about the upper bounds in virtual weight restricted AES Method will depend on the degree of the DM's allowance for stressing strengths and 'covering up' weaknesses.

**Table 19. AES ranking of the first 20 programs with virtual weight restrictions (+25%)**

<b>Rank</b>	<b>Program</b>	<b>Area - Virtual (+25%, No LB)</b>	<b>Dk</b>	<b>Efficiency scores with no deletion</b>
1	1	0.0000	0	1.0000
1	2	0.0000	0	1.0000
1	3	0.0000	0	1.0000
1	5	0.0000	0	1.0000
2	4	0.0522	3	0.9673
3	6	0.1099	4	0.9530
4	7	0.2418	3	0.8903
5	9	0.2697	5	0.9099
6	8	0.3624	6	0.9133
7	10	0.4976	7	0.8786
8	13	0.6237	7	0.8520
9	11	0.8326	8	0.8441
10	14	0.8396	9	0.8404
11	15	0.8470	10	0.8487
12	16	0.9375	10	0.8442
13	12	0.9560	10	0.8273
14	18	1.1717	12	0.8058
15	20	1.1900	13	0.8177
16	19	1.2249	11	0.8126
17	17	1.5346	11	0.7497

**Table 20. AES ranking of the first 20 programs with virtual weight restrictions (+40%)**

<b>Rank</b>	<b>Program</b>	<b>Area - Virtual (+40%, No LB)</b>	<b>Dk</b>	<b>Efficiency scores with no deletion</b>
1	1	0.0000	0	1.0000
1	2	0.0000	0	1.0000
1	3	0.0000	0	1.0000
1	5	0.0000	0	1.0000
1	7	0.0000	0	1.0000
2	4	0.0036	1	0.9964
3	6	0.0044	1	0.9956
4	9	0.0577	2	0.9658
5	19	0.1397	4	0.9375
6	8	0.1767	5	0.9399
7	11	0.2179	5	0.9162
8	17	0.3278	7	0.9109
9	10	0.3437	5	0.8939
10	13	0.4065	5	0.8669
11	15	0.5247	6	0.8746
12	12	0.5823	8	0.8529
13	18	0.5923	7	0.8544
14	14	0.6435	8	0.8527
15	16	0.7535	9	0.8569
16	20	0.9096	10	0.8366

## Assurance Regions in the AES Method for the MBA Ranking Problem

Assurance Regions (AR) by Thompson et al. (1990) are for including restrictions in DEA by specifying relative values of the inputs/outputs. These specifications are incorporated into DEA models by imposing direct restrictions on the weights of inputs and outputs. AR-I type of restrictions are for use among inputs or outputs only; AR-II type of restrictions can be used also for relating inputs and outputs. As our case of MBA ranking includes only outputs (except for the artificial input of 1 we assigned), we can utilize AR-I type of restrictions.

FT provides us with weights for the outputs in the form of percentages. By relating these percentages to each other, we can arrive at relative values for the outputs' weights. First we need to choose a reference output, let Output 1 be the reference. Then we can represent each weight as a multiple of Output 1's weight. As relative values expressed in equalities will be too strict and AR is usually applied in ranges, again  $\mp 25\%$  and  $\mp 40\%$  ranges will be constructed. For  $u_i$  being the weight of output  $i$ , and Output 1 being the reference output; the lower and upper bounds of the weights for outputs 2, ..., 20 for  $\mp 25\%$  and  $\mp 40\%$  ranges will be as in Table 21.

**Table 21.**  $\mp 25\%$  and  $\mp 40\%$  ranges for the weights in the AR approach

Weight	$\mp 25\%$		$\mp 40\%$	
	LB	UB	LB	UB
$u_2$	$0.7500 u_1$	$1.2500 u_1$	$0.6000 u_1$	$1.4000 u_1$
$u_3$	$0.1125 u_1$	$0.1875 u_1$	$0.0900 u_1$	$0.2100 u_1$
$u_4$	$0.1125 u_1$	$0.1875 u_1$	$0.0900 u_1$	$0.2100 u_1$
$u_5$	$0.1125 u_1$	$0.1875 u_1$	$0.0900 u_1$	$0.2100 u_1$
$u_6$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_7$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_8$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_9$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_{10}$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_{11}$	$0.0375 u_1$	$0.0625 u_1$	$0.0300 u_1$	$0.0700 u_1$
$u_{12}$	$0.1500 u_1$	$0.2500 u_1$	$0.1200 u_1$	$0.2800 u_1$
$u_{13}$	$0.1500 u_1$	$0.2500 u_1$	$0.1200 u_1$	$0.2800 u_1$
$u_{14}$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_{15}$	$0.2250 u_1$	$0.3750 u_1$	$0.1800 u_1$	$0.4200 u_1$
$u_{16}$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_{17}$	$0.0750 u_1$	$0.1250 u_1$	$0.0600 u_1$	$0.1400 u_1$
$u_{18}$	$0.1875 u_1$	$0.3125 u_1$	$0.1500 u_1$	$0.3500 u_1$
$u_{19}$	$0.1875 u_1$	$0.3125 u_1$	$0.1500 u_1$	$0.3500 u_1$
$u_{20}$	$0.3750 u_1$	$0.6250 u_1$	$0.3000 u_1$	$0.7000 u_1$

Note that by relating all weights to weight 1, we relate every weight to each other. Table 22 shows the AES ranking of the 20 programs with AR weight restrictions, with  $\mp 25\%$  and  $\mp 40\%$  ranges. One can see that unlike the case of virtual weight restrictions, wider ranges in AR does not lead to an increased number of programs with areas of 0. Also there are no programs which changed its rank position greatly (like Program 19 with virtual weight restrictions). Areas get smaller with  $\mp 40\%$  ranges but the difference is not as significant as in virtual weight restrictions. The average decrease in area values when we switch to a wider range is 0.3004 with virtual weight restrictions whereas it is 0.1629 with AR. To check whether virtual weight restrictions are more sensitive to range changes than AR, let us study the changes in weights in both approaches between 25 and 40% ranges. In virtual weight restrictions, as the constraints are on weighted outputs in the form of percentages, the change in these percentages must be studied. For example, the change in the percentage of weighted output 1 in the sum of all weighted outputs must be examined. For AR, to make the two approaches comparable, the change in the percentage of individual weights in the sum of all weights will be studied. Of course, the weights are DMU specific, so the changes can be studied on one DMU at a time. For both approaches, the changes for all weights of a DMU will be taken in absolute values and averaged. These average absolute changes constitute the final measures for the sensitivity of the two approaches to different widths of ranges used. Obviously one can study on as many DMUs as s/he likes and consider all results together. Table 23 shows the changes in the weights of 3 example DMUs between 25 and 40% ranges with the two approaches. We can see that the virtual weight restrictions resulted in more change in the weights of all three programs when the ranges are altered. The results signal that virtual weight restrictions are more sensitive to changes made in the ranges.

Both virtual weight restrictions and AR can be used with the AES Method. We choose to employ virtual weight restrictions for the ranking problem of all programs because our results with 20 programs point to increased sensitivity of this approach to changes in the ranges. This sensitivity may be useful in obtaining different rankings with different DM specifications. In addition, the weights provided by FT are more suited to be converted to virtual weight restrictions as they are already in the form of percentages.



**Table 22. AES ranking of the first 20 programs with AR ( $\mp$  25% and  $\mp$  40%)**

$\mp$ 25%			$\mp$ 40%		
Rank	Program	Area – AR ( $\mp$ 25%)	Rank	Program	Area – AR ( $\mp$ 40%)
1	1	0.0000	1	1	0.0000
1	2	0.0000	1	2	0.0000
1	3	0.0000	1	3	0.0000
1	5	0.0000	1	5	0.0000
1	7	0.0000	1	7	0.0000
2	4	0.0317	2	4	0.0005
3	6	0.1164	3	6	0.0615
4	9	0.1946	4	9	0.0758
5	8	0.3149	5	11	0.1154
6	11	0.3462	6	8	0.1884
7	10	0.4959	7	10	0.3547
8	13	0.7345	8	19	0.4019
9	12	0.7460	9	17	0.5124
10	17	0.8141	10	12	0.5205
11	19	0.8513	11	13	0.5221
12	15	0.8633	12	14	0.6515
13	14	0.8778	13	18	0.6574
14	18	1.0051	14	15	0.6607
15	16	1.0197	15	16	0.8059
16	20	1.5857	16	20	1.2100

**Table 23. Changes in weights with virtual weight restrictions and AR (25-40% ranges)**

Program	Change in weights (25-40%)	
	Virtual Weight Restrictions	Assurance Regions
4	0.0129	0.0066
10	0.0177	0.0072
19	0.0107	0.0081

### 5.3. The Results of the MBA Ranking Problem with the AES Method

In this section, we provide the experimental results of our AES Method on the MBA program ranking problem. We will rank the 81 MBA programs that Köksalan et al. provide scores for in twenty criteria. As mentioned in the previous section, all programs are assumed to have a single input of 1. Virtual weight restrictions are used with the AES Method. Using the fixed weights the FT provides, only upper bounds are constructed for the virtual outputs. Both +25% and +40% upper bounds are used in the experiments. Table 24 shows the ranking results of the AES Method (with both +25% and +40% upper bounds), programs' area values and additionally their original efficiency scores and  $D_k$  values.

When we look at the results, we see that the AES rankings follow neither the  $D_k$  nor the efficiency score order. Obviously, programs with 0  $D_k$  values and 1 efficiency scores are the programs with areas of 0 and they share the 1<sup>st</sup> rank. Comparing the +25 and 40% upper bounds, we see that +40% allows programs more flexibility which results in lower  $D_k$  values and higher efficiency scores. The number of programs sharing the 1<sup>st</sup> rank is also higher. When we look down the list for +25% upper bounds, we see that 3 programs (57, 71 and 81) cannot reach efficiency no matter how many programs leave the set, so they do not have  $D_k$  values. Their original efficiency scores are also 0 which show significant inefficiency. When we observe their output values from Table 16, we see that they have 0 values in a number of outputs whose weights (original FT weights) sum up to big percentages. Program 57 has 0 values in outputs totaling 22%, Program 71 again 22% and Program 81 31%. These percentages are the highest among the 81 program set and prevent the three mentioned programs reaching efficiency. Within the weight restrictions imposed, they cannot offset their 0's by their strengths. When the upper bounds are constructed by +40%, only Program 81, which has the highest percentage of 0's, remains in this situation.

We have made several discussions in previous chapters that the use of  $D_k$  values or classical efficiency scores may result in misleading outcomes. We have proposed that the use of the area values of the AES Method will prevent such outcomes. Now, let us confirm our propositions with the help of results presented in Table 24.

**Table 24. Results of the MBA ranking problem by the AES Method**

+25% Upper Bounds					+40% Upper Bounds				
Rank	Program	Area	D <sub>k</sub>	Eff. Score	Rank	Program	Area	D <sub>k</sub>	Eff. Score
1	1	0.0000	0	1.0000	1	1	0.0000	0	1.0000
1	2	0.0000	0	1.0000	1	2	0.0000	0	1.0000
1	3	0.0000	0	1.0000	1	3	0.0000	0	1.0000
1	5	0.0000	0	1.0000	1	5	0.0000	0	1.0000
2	4	0.0520	3	0.9674	1	7	0.0000	0	1.0000
3	6	0.1098	4	0.9531	2	4	0.0036	1	0.9964
4	7	0.2424	2	0.8901	3	6	0.0048	1	0.9953
5	9	0.2700	5	0.9098	4	9	0.0616	3	0.9641
6	8	0.3311	5	0.9134	5	19	0.1582	5	0.9349
7	10	0.4816	7	0.8786	6	8	0.1796	5	0.9397
8	13	0.6364	7	0.8520	7	11	0.2368	5	0.9149
9	14	0.8036	9	0.8405	8	10	0.3434	5	0.8940
10	11	0.8318	8	0.8443	9	17	0.3476	7	0.9096
11	15	0.8472	10	0.8487	10	13	0.4067	5	0.8669
12	16	0.9368	10	0.8443	11	15	0.5295	6	0.8738
13	12	0.9761	10	0.8273	12	12	0.5823	8	0.8529
14	20	1.2044	14	0.8178	13	18	0.6217	8	0.8536
15	18	1.2419	13	0.8058	14	14	0.6438	8	0.8527
16	19	1.3429	13	0.8123	15	16	0.7544	9	0.8570
17	21	1.6993	18	0.7904	16	20	0.9052	10	0.8366
18	23	1.9085	18	0.7574	17	21	1.0184	10	0.8192
19	28	2.1760	21	0.7635	18	26	1.2168	14	0.8393
20	24	2.2287	19	0.7459	19	23	1.3816	13	0.7914
21	25	2.4236	19	0.7279	20	24	1.4981	16	0.7811
22	17	2.7232	24	0.7501	21	28	1.5261	16	0.7945
23	22	2.8938	21	0.7258	22	25	1.7404	16	0.7676
24	27	2.9368	22	0.7183	23	35	1.9008	16	0.7321
25	31	3.2170	21	0.7084	24	33	2.0148	22	0.7648
26	32	3.2444	24	0.7107	25	22	2.0169	16	0.7637
27	37	3.2668	25	0.6983	26	27	2.2351	17	0.7401
28	38	3.5297	27	0.6884	27	39	2.2438	27	0.7635
29	26	3.5365	27	0.7208	28	32	2.3333	18	0.7470
30	29	3.6839	26	0.6753	29	30	2.3770	22	0.7788
31	33	3.8382	28	0.7031	30	37	2.4284	21	0.7330
32	42	4.3604	31	0.6924	31	34	2.4552	22	0.7360
33	30	4.5087	31	0.6886	32	29	2.4772	23	0.7417
34	35	4.5251	28	0.6558	33	31	2.5880	19	0.7367
35	43	5.0166	34	0.6507	34	38	2.8994	22	0.7214
36	34	5.2558	34	0.6560	35	42	2.9096	29	0.7572
37	41	5.2709	34	0.6275	36	41	3.1486	22	0.6866
38	48	5.7735	35	0.6359	37	45	3.2190	21	0.6617

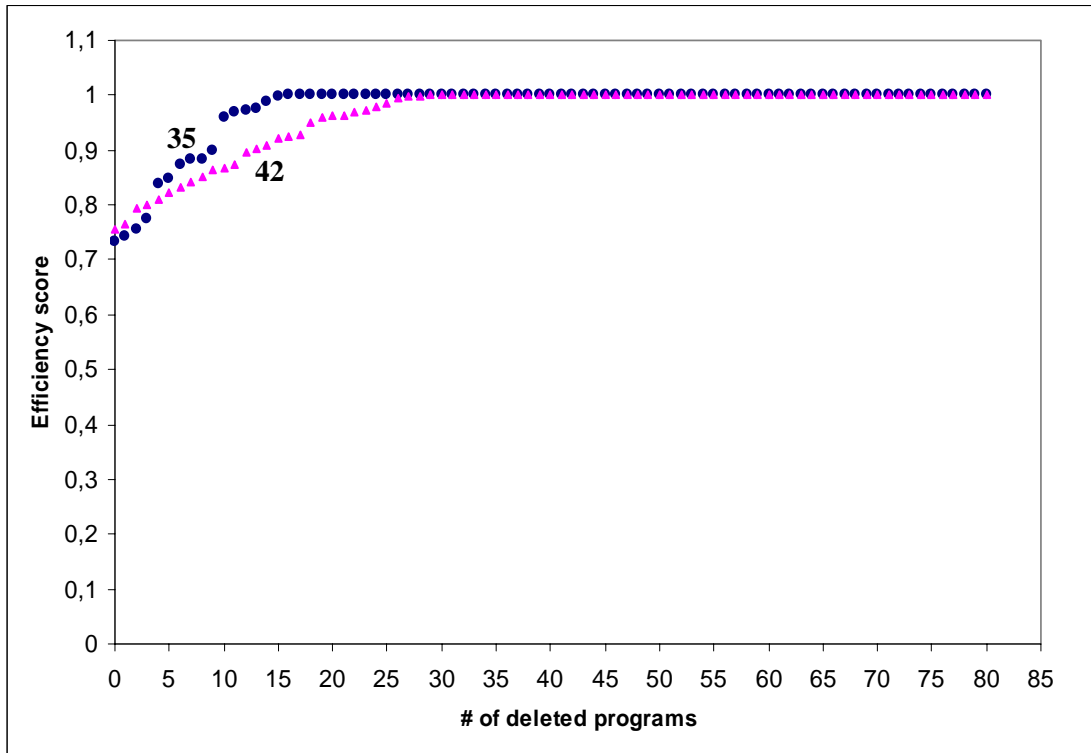
**Table 24 (continued)**

+25% Upper Bounds					+40% Upper Bounds				
Rank	Program	Area	D <sub>k</sub>	Eff. Score	Rank	Program	Area	D <sub>k</sub>	Eff. Score
39	40	5.9238	37	0.6602	38	43	3.3933	28	0.7195
40	45	6.0727	38	0.6119	39	40	3.7331	31	0.7319
41	46	7.2048	42	0.6216	40	36	4.0624	31	0.7105
42	51	7.2092	43	0.6294	41	47	4.3801	38	0.6835
43	55	7.4624	42	0.6181	42	48	4.4489	27	0.6476
44	47	7.4941	43	0.5836	43	44	4.5200	32	0.6590
45	36	7.5374	42	0.6122	44	46	4.7113	32	0.6765
46	56	7.7234	43	0.5938	45	51	4.9203	35	0.6747
47	54	7.7532	45	0.6116	46	55	5.7289	31	0.6397
48	44	8.0998	43	0.6015	47	49	5.7733	38	0.6592
49	58	8.8185	48	0.5640	48	53	5.8288	36	0.6269
50	53	9.0823	48	0.5665	49	56	6.1282	36	0.6457
51	52	9.3479	46	0.5305	50	54	6.2538	35	0.6376
52	49	9.5800	48	0.5970	51	52	6.7398	38	0.6078
53	39	9.7934	50	0.5631	52	58	7.0681	42	0.6134
54	59	10.3740	50	0.5561	53	68	8.0767	42	0.5461
55	61	12.4780	55	0.5488	54	59	8.1744	42	0.5944
56	60	14.1330	60	0.5240	55	61	9.1634	50	0.5815
57	64	14.6750	57	0.4955	56	67	9.3177	53	0.5849
58	50	15.8650	58	0.4327	57	60	10.8460	52	0.5626
59	63	16.0150	62	0.4647	58	64	11.9080	55	0.5330
60	66	16.2960	60	0.4548	59	63	11.9610	57	0.5378
61	62	16.9780	63	0.4822	60	50	12.3470	53	0.4890
62	69	17.1290	63	0.4929	61	66	12.7610	52	0.4963
63	65	18.3460	64	0.4401	62	62	14.1770	59	0.5144
64	72	19.0180	63	0.4341	63	69	14.6260	59	0.5368
65	73	26.7170	74	0.3930	64	72	14.6330	57	0.4761
66	78	27.4280	71	0.3958	65	65	14.7210	60	0.5161
67	70	27.6650	71	0.3466	66	79	17.7200	61	0.4079
68	80	29.0360	71	0.3498	67	76	17.8640	63	0.3936
69	76	29.6970	73	0.3059	68	70	18.8630	64	0.4178
70	68	34.4900	76	0.3488	69	73	22.3390	70	0.4456
71	75	35.0220	75	0.3215	70	78	22.8850	69	0.4228
72	77	35.6200	76	0.2924	71	75	22.8950	69	0.3696
73	67	43.8230	76	0.2058	72	80	24.0910	69	0.4108
74	79	50.0900	79	0.1888	73	71	24.3050	69	0.3694
75	74	51.7110	79	0.1491	74	77	28.7290	75	0.3226
76	71	80.9850	-	0.0002	75	57	32.9220	75	0.3551
77	57	81.0000	-	0.0000	76	74	41.5450	79	0.2304
78	81	81.0000	-	0.0000	77	81	81.0000	-	0.0000

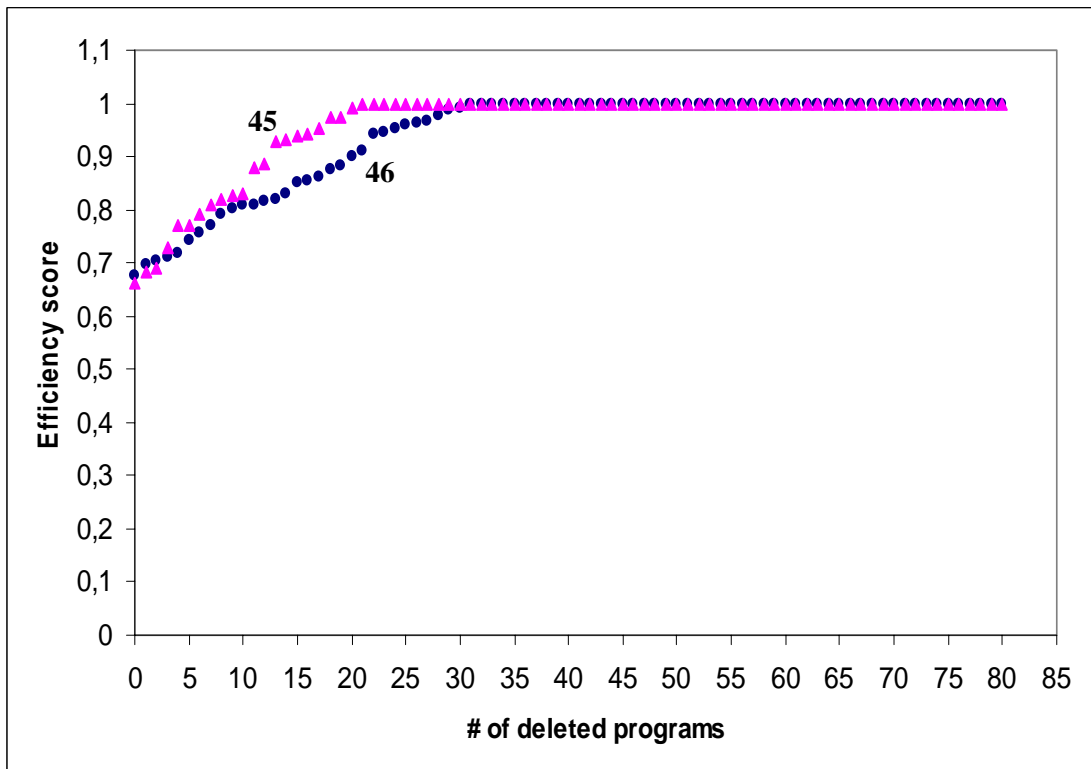
Let us choose pairs of programs from our set in such a way that the relative ranking of the AES Method does not agree with the relative ranking of  $D_k$  or the classical efficiency score. In other words, among the two programs on hand, the AES method will place one higher whereas the  $D_k$  or the classical efficiency score will place the other. If we produce the programs' Efficiency score - # of deleted DMUs graph, we can observe the reasons of the differences. +40% upper bounds will be used in all illustrations here. This choice is arbitrary and has no effect on the discussion.

We will first look at ranking differences of AES and the classical efficiency score. Let programs 35 and 42 be our first pair. As can be checked from Table 24, AES places Program 35 twelve ranks higher than 42. However, as the classical efficiency score of 42 is higher than 35; efficiency score rank of Program 42 will be higher. Figure 12 illustrates the Efficiency score - # of deleted programs graph of the two programs. We can see that although Program 35 starts with a lower efficiency score, it is a much quicker improver than Program 42. This quick improvement is realized and rewarded in the AES Method. The pair of programs 45 and 46 exhibits the same behavior. The AES Method places Program 45 seven ranks higher than 46. However, as its original efficiency score is lower, it will be ranked lower if the classical efficiency scores are used. Figure 13 illustrates the graph of this second pair; one can see the quick improvement of Program 45.

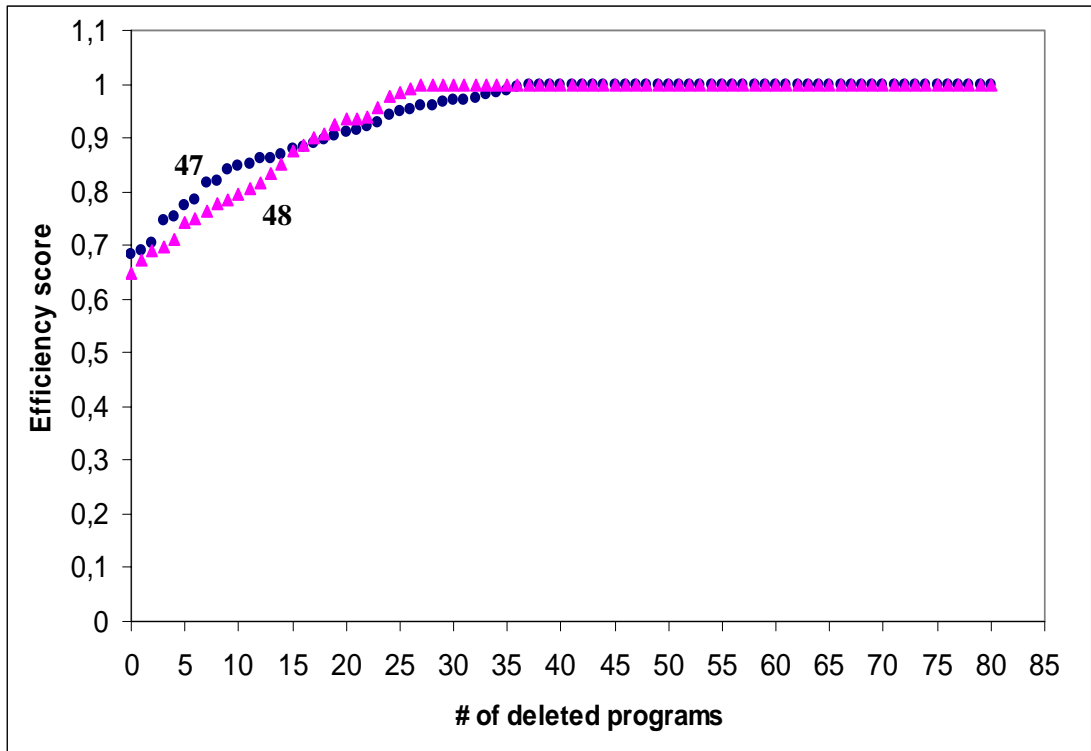
Now let us look at ranking differences between the AES Method and the  $D_k$ . In the pair of programs 47 and 48, 47 is ranked 1 place higher than 48 with the AES Method, however the  $D_k$  value of 48 is 11 less than the  $D_k$  value of 47. When we look at Figure 14, we see that despite its high  $D_k$ , Program 47 manages to sustain high efficiency scores, higher than Program 48 for a sufficiently long time to be placed higher. The property of maintaining high efficiency scores is more evident in the pair of programs 31 and 39. Program 31 has a much less  $D_k$  value than Program 39 (19 as opposed to 27) but the AES Method places 39 six ranks higher. The reason can be observed from Figure 15; Program 39's curve lies above the curve of 31 for most values for the number of deleted programs. The reason for Program 39's high  $D_k$  value is the problem set characteristics. AES can detect this and make its ranking accordingly.



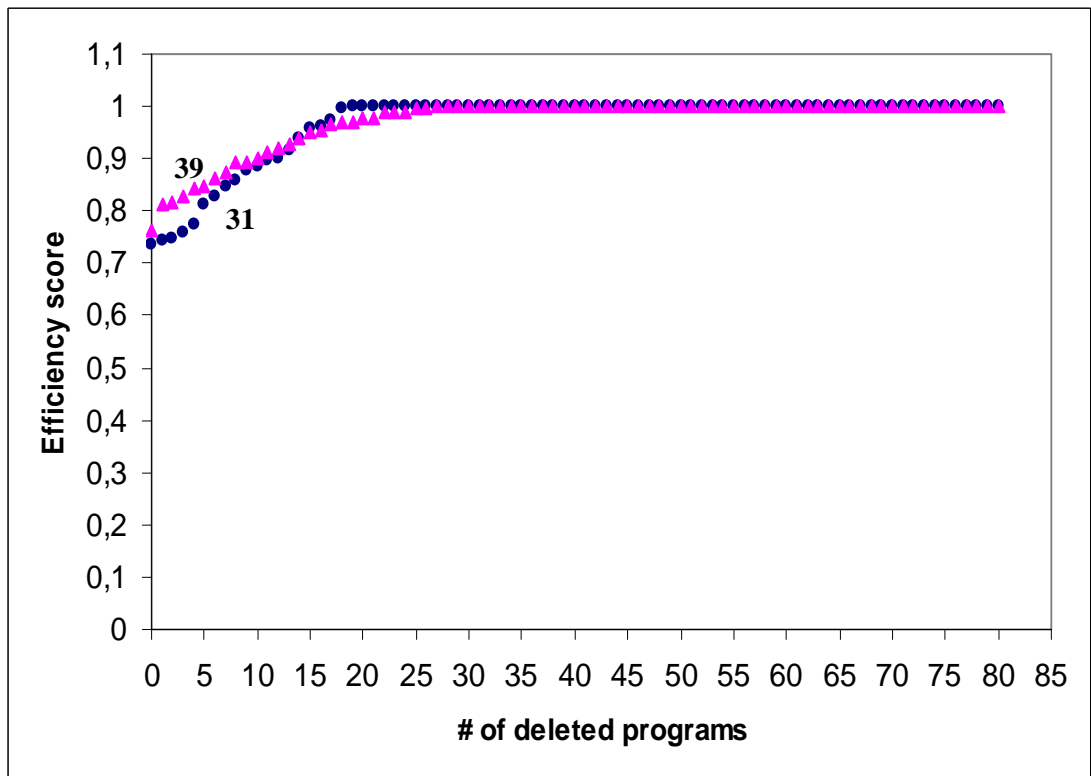
**Figure 12.** The Efficiency score - # of deleted programs graph of programs 35 and 42



**Figure 13.** The Efficiency score - # of deleted programs graph of programs 45 and 46



**Figure 14.** The Efficiency score - # of deleted programs graph of programs 47 and 48



**Figure 15.** The Efficiency score - # of deleted programs graph of programs 31 and 39

Besides these four pairs of programs we used to illustrate the ranking differences of AES and classical efficiency score, and also of AES and  $D_k$ , there are also interesting cases where AES neither agrees with efficiency score nor  $D_k$ . Consider programs 11 and 14. Efficiency score of 11 is higher than 14 (0.8318 as opposed to 0.8036) and its  $D_k$  value is also lower (8 as opposed to 9); but AES places Program 14 one rank higher than 11. Note that this condition will require the score curves of the programs to intersect more than once. If a program starts with a lower efficiency score, finishes with a higher  $D_k$  value and still has a lower area than the other, it should necessarily raise its score curve above the one of other at some point in the graph and lower it eventually to reach efficiency later. Let us illustrate this case with the graphical representation of a new pair of programs, 31 and 34. Although 31 has a higher efficiency score and a lower  $D_k$ , it is placed 2 ranks lower than 34 with AES. Figure 16 shows the score change graph of the two programs; y-axis shows scores above the starting efficiency scores in order to make score curves' behaviors more easily observable. Program 34 is ranked higher with AES since it manages to keep its scores higher for long enough to make its area smaller.

With these illustrations, we have shown that the MBA ranking of AES avoids shortcomings of ranking by  $D_k$  values and classical efficiency scores.

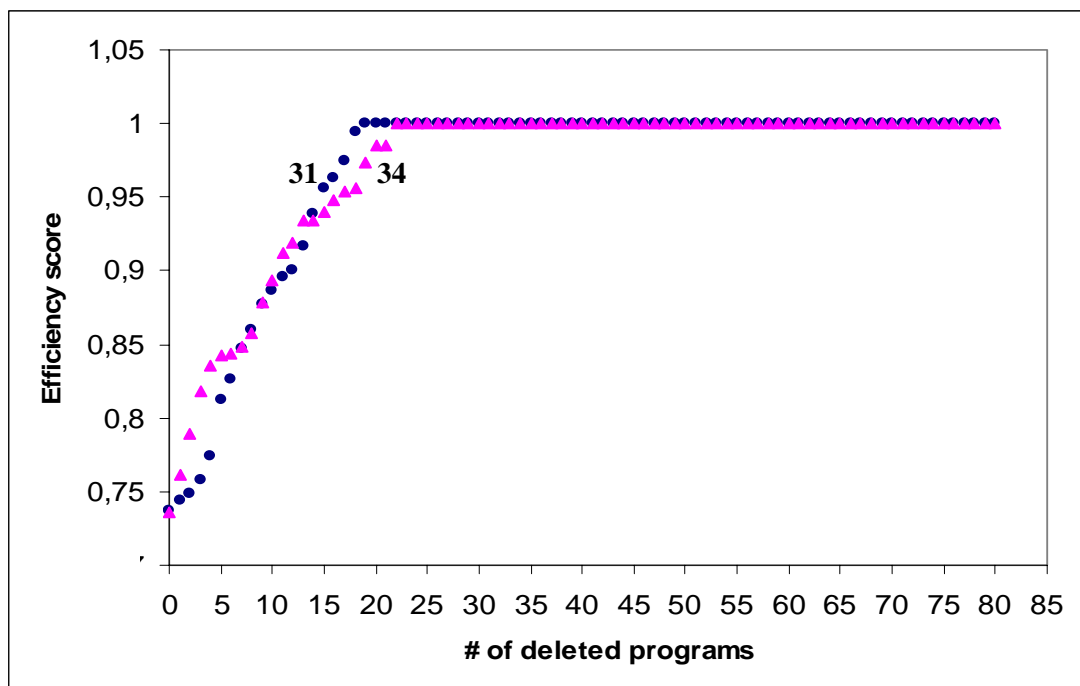


Figure 16. The Efficiency score - # of deleted programs graph of programs 31 and 34



#### **5.4. The Results of the MBA Ranking Problem with the Cross-Efficiency Method**

In this section, we provide the results of the MBA Ranking problem with the Cross-Efficiency Method to observe the differences from the AES Method. In the Cross-Efficiency Method, DMUs are ranked according to their Cross-Efficiency scores which are averaged values of their efficiency scores calculated with the weights of all DMUs. For our problem, weight restrictions will be included in Cross-Efficiency Analysis to take preferences into consideration and to enable comparison. These restrictions will again be in the form of virtual weight restrictions. Only upper bounds will be constructed by +25% and +40% increased percentages of FT. So, the weights used in the Cross-Efficiency Method will be the weights in the AES Method when no programs are removed from the set yet.

Table 25 shows the results of the MBA Ranking Problem by the Cross-Efficiency Method. Programs' AES ranks are also included to reveal the differences. When studying the differences, it should be kept in mind that a number of programs share the first rank in the AES Method. It can be observed that the two rankings are quite different from each other.

Let us again study pairs of programs; we choose pairs in such a way that the AES and Cross-Efficiency methods rank different programs higher. We arbitrarily choose +40% upper bounds case. Programs 50 and 68 being our first pair, Figure 17 exhibits their Efficiency score – number of deleted programs graph. We can see that Program 68's curve always lies above the one of 50, so it is definitely superior to 50. AES Method places Program 68 seven ranks higher; however, Cross-Efficiency Method places Program 50 fourteen ranks higher than 68. Our second pair, Programs 14 and 17 in Figure 18, exhibit the same situation. AES Method places Program 17 five ranks higher as its curve is always higher than the one of Program 14. However, Cross-Efficiency ranks Program 14 seven places higher than 17.

We conclude that the Cross-Efficiency Method and the AES Method produce different ranking lists and the former method does not take the issues considered by the AES into account. A DMU may sustain higher efficiency scores (with different numbers of deleted DMUs) than another, it may be a quick improver, but it still may be ranked lower by the Cross-Efficiency Analysis. This approach is more dependent on how much a DMU can utilize the weights of the other DMUs.

**Table 25. Results of the MBA ranking problem by Cross-Efficiency Method**

+25% Upper Bounds				+40% Upper Bounds			
Rank	Program	Cross-Efficiency Score	AES Rank	Rank	Program	Cross-Efficiency Score	AES Rank
1	1	0.9509	1	1	1	0.9690	1
2	2	0.9494	1	2	2	0.9672	1
3	3	0.9423	1	3	3	0.9634	1
4	5	0.9182	1	4	5	0.9469	1
5	4	0.9139	2	5	4	0.9310	2
6	6	0.8886	3	6	6	0.9038	3
7	7	0.8614	4	7	7	0.8917	1
8	9	0.8541	5	8	9	0.8780	4
9	8	0.8296	6	9	8	0.8457	6
10	10	0.8086	7	10	11	0.8311	7
11	11	0.7940	10	11	10	0.8253	8
12	14	0.7784	9	12	14	0.8006	14
13	15	0.7747	11	13	13	0.7956	10
14	13	0.7721	8	14	15	0.7950	11
15	12	0.7696	13	15	19	0.7867	5
16	16	0.7650	12	16	12	0.7849	12
17	17	0.7558	22	17	16	0.7841	15
18	18	0.7469	15	18	18	0.7807	13
19	19	0.7447	16	19	17	0.7787	9
20	21	0.7342	17	20	21	0.7709	17
21	20	0.7276	14	21	24	0.7541	20
22	24	0.7185	20	22	20	0.7516	16
23	23	0.7090	18	23	23	0.7367	19
24	22	0.7007	23	24	30	0.7185	29
25	28	0.6849	19	25	22	0.7151	25
26	26	0.6835	29	26	28	0.7128	21
27	25	0.6801	21	27	26	0.7124	18
28	30	0.6780	33	28	25	0.7077	22
29	37	0.6644	27	29	34	0.7008	31
30	27	0.6626	24	30	29	0.6963	32
31	29	0.6626	30	31	33	0.6942	24
32	33	0.6617	31	32	35	0.6933	23
33	31	0.6596	25	33	37	0.6931	30
34	35	0.6586	34	34	32	0.6822	28
35	34	0.6578	36	35	27	0.6776	26
36	32	0.6513	26	36	31	0.6774	33
37	38	0.6497	28	37	38	0.6760	34
38	36	0.6342	45	38	36	0.6691	40
39	43	0.6219	35	39	39	0.6640	27
40	39	0.6216	53	40	45	0.6510	37
41	45	0.6163	40	41	43	0.6501	38

**Table 25 (continued)**

+25% Upper Bounds				+40% Upper Bounds			
Rank	Program	Cross-Efficiency Score	AES Rank	Rank	Program	Cross-Efficiency Score	AES Rank
42	42	0.6133	32	42	47	0.6459	41
43	41	0.6111	37	43	42	0.6434	35
44	47	0.6100	44	44	49	0.6391	47
45	49	0.6093	52	45	41	0.6362	36
46	40	0.6071	39	46	40	0.6336	39
47	46	0.6058	41	47	46	0.6310	44
48	44	0.6009	48	48	44	0.6309	43
49	53	0.5992	50	49	53	0.6278	48
50	48	0.5906	38	50	51	0.6120	45
51	51	0.5784	42	51	48	0.6063	42
52	50	0.5669	58	52	50	0.5969	60
53	52	0.5659	51	53	52	0.5957	51
54	54	0.5596	47	54	54	0.5866	50
55	55	0.5542	43	55	57	0.5847	75
56	57	0.5512	77	56	55	0.5752	46
57	56	0.5457	46	57	56	0.5687	49
58	58	0.5428	49	58	58	0.5686	52
59	59	0.5366	54	59	62	0.5648	62
60	60	0.5290	56	60	59	0.5603	54
61	62	0.5264	61	61	60	0.5597	57
62	67	0.5253	73	62	61	0.5512	55
63	61	0.5238	55	63	67	0.5436	56
64	65	0.5055	63	64	65	0.5368	65
65	63	0.5019	59	65	63	0.5228	59
66	68	0.4924	70	66	68	0.5141	53
67	64	0.4773	57	67	64	0.4980	58
68	66	0.4623	60	68	77	0.4933	74
69	69	0.4614	62	69	74	0.4931	76
70	71	0.4608	76	70	71	0.4929	73
71	70	0.4564	67	71	70	0.4924	68
72	74	0.4551	75	72	69	0.4872	63
73	77	0.4503	72	73	66	0.4855	61
74	72	0.4491	64	74	72	0.4740	64
75	73	0.4216	65	75	73	0.4544	69
76	79	0.4173	74	76	79	0.4533	66
77	76	0.4141	69	77	76	0.4489	67
78	78	0.4083	66	78	75	0.4357	71
79	80	0.4037	68	79	80	0.4323	72
80	75	0.4026	71	80	78	0.4307	70
81	81	0.3903	78	81	81	0.4224	77

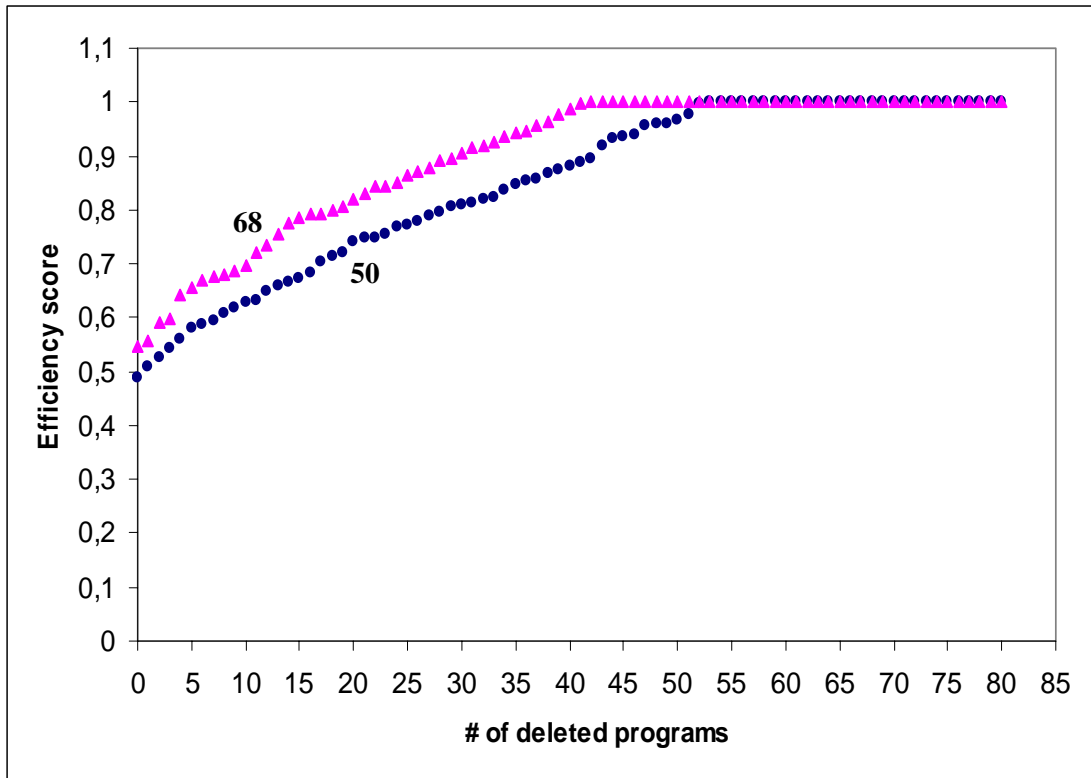


Figure 17. The Efficiency score - # of deleted programs graph of programs 50 and 68

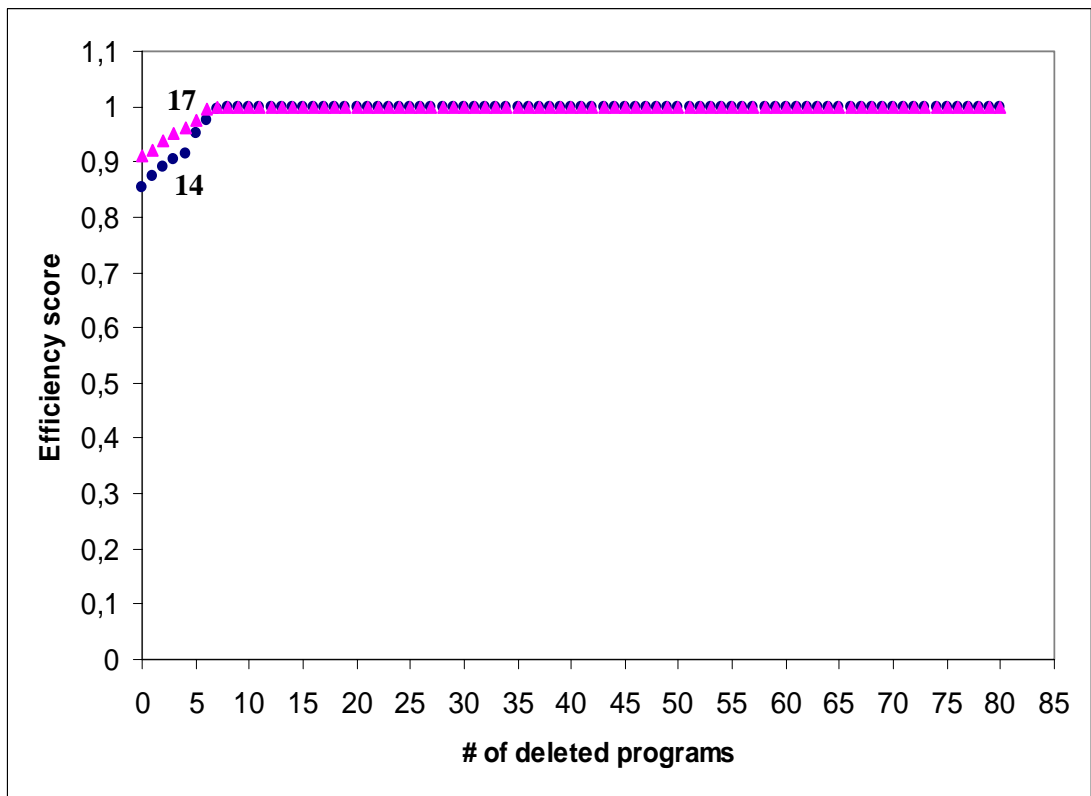


Figure 18. The Efficiency score - # of deleted programs graph of programs 14 and 17

Now that we have ranked the programs by +25 and 40% ranged AES and Cross-Efficiency Method, we can study rank correlations between different ranges and methods. Using Spearman rank correlation coefficient (Spearman, 1904), Table 26 shows the correlations between pairs of rankings we make from AES, Cross-Efficiency and FT rankings. It also includes average absolute rank differences between them. Table A in Appendix shows the ranks of the programs according to AES and Cross-Efficiency with both ranges in the rank order of FT, and also rank differences of all programs with different pairs of lists. Note that in Table A, programs sharing the first rank with AES are all assigned the mean of the rank positions for which they are tied. This is necessary for the computation of coefficients and for the comparison of the lists.

When we study Table 26, we see that the correlation coefficients are high for all pairs of rank lists. However, when we examine Table A, we notice rank differences as high as 25 despite high correlation coefficients. The maximum absolute rank differences are above 20 for all comparisons including AES (which result in correlation coefficients as high as 0.9792); whereas the maximum absolute rank differences between two ranges of Cross-Efficiency and between Cross-Efficiency and FT are below 10 (which result in coefficients as high as 0.9972). Considering Spearman rank correlation coefficients, and also the average and maximum rank differences, we can conclude on higher sensitivity of AES to range changes as opposed to Cross-Efficiency, and also higher similarity of Cross-Efficiency to FT ranking as opposed to AES.

**Table 26. Spearman rank correlation coefficients and average rank differences**

	<b>Spearman rank correlation coefficient</b>	<b>Average absolute rank difference</b>
<b>AES +25 – 40%</b>	0.9735	3.6296
<b>CROSS +25 – 40%</b>	0.9972	1.0617
<b>AES +25% - CROSS +25%</b>	0.9690	4.1728
<b>AES +40%- CROSS +40%</b>	0.9754	3.7284
<b>AES +25% - FT</b>	0.9727	3.8272
<b>AES +40% - FT</b>	0.9792	3.2346
<b>CROSS +25% - FT</b>	0.9953	1.6049
<b>CROSS +40% - FT</b>	0.9921	2.2222

## 5.5. Sorting MBA Programs

Our proposed method AES produces scores for the DMUs for the main purpose of ranking. However, DMs may demand ranking of classes of DMUs. Moreover, some DMUs' scores may not be significantly different from each other to enable a strict ranking list. As a result, sorting of the DMUs may be required. In this section, we provide results of sorting programs by their AES scores. Programs are also sorted by their Cross-Efficiency scores to provide the outcomes of an alternative method.

As mentioned with the discussion of the AES Method, an incremental clustering algorithm can be used on the AES scores of the programs. We provided the algorithm that we will employ here in Chapter 4. Before the application, though, we need to determine the threshold value we will use to decide on the classes. But let us first study the AES scores of the programs once more from Table 24. For both cases of +25% and +40% upper bounds, the AES scores of the programs become more distant from each other as we move down the ranking list. The differences between the scores are in decimal places in the beginning, however they get larger and larger towards the end. As a result, if we use a single threshold for sorting, upper classes will be denser than the lower ones, which will include very few programs or even sometimes only one program. Programs placed lower should not be discriminated by the same threshold used for upper placed programs. To decide on which point to change the threshold, let us study the distribution of the AES scores. Figure 19 shows the AES scores of the programs in rank order for the case of +25% upper bounds. Figure 20 performs the same task for the +40% case. We can see the increasing slope of the AES scores in the graphs. We will change the threshold value at points in the graphs where the slope makes a significant change. These points are illustrated by cut-off lines in the figures. These cut-off points are right after programs 39 and 72 for the +25% case; and 58 and 71 for the +40% case.

Accordingly, we will use three threshold values for the incremental clustering algorithm. We will start with one, increase it when we come to the first cut-off point and increase it again when we hit the second cut-off point. The decision on the thresholds depends on the number of classes desired; large thresholds will result in few classes, smaller thresholds will result in more.

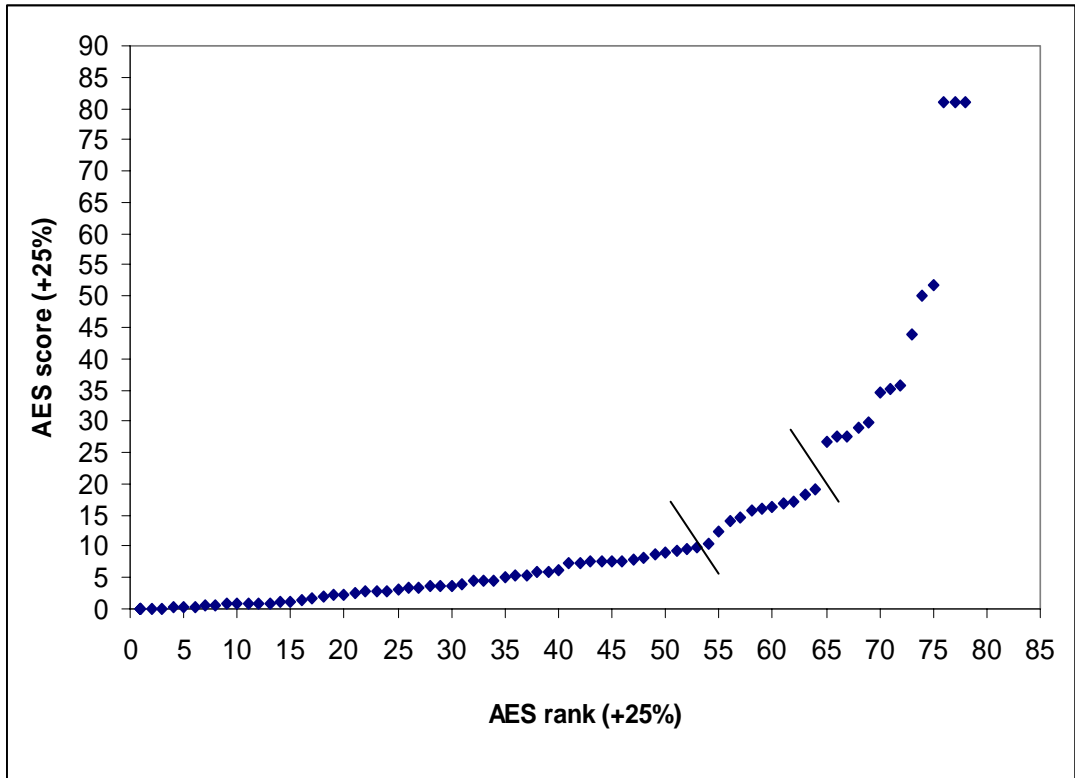


Figure 19. The AES score distribution of the programs for +25% upper bounds

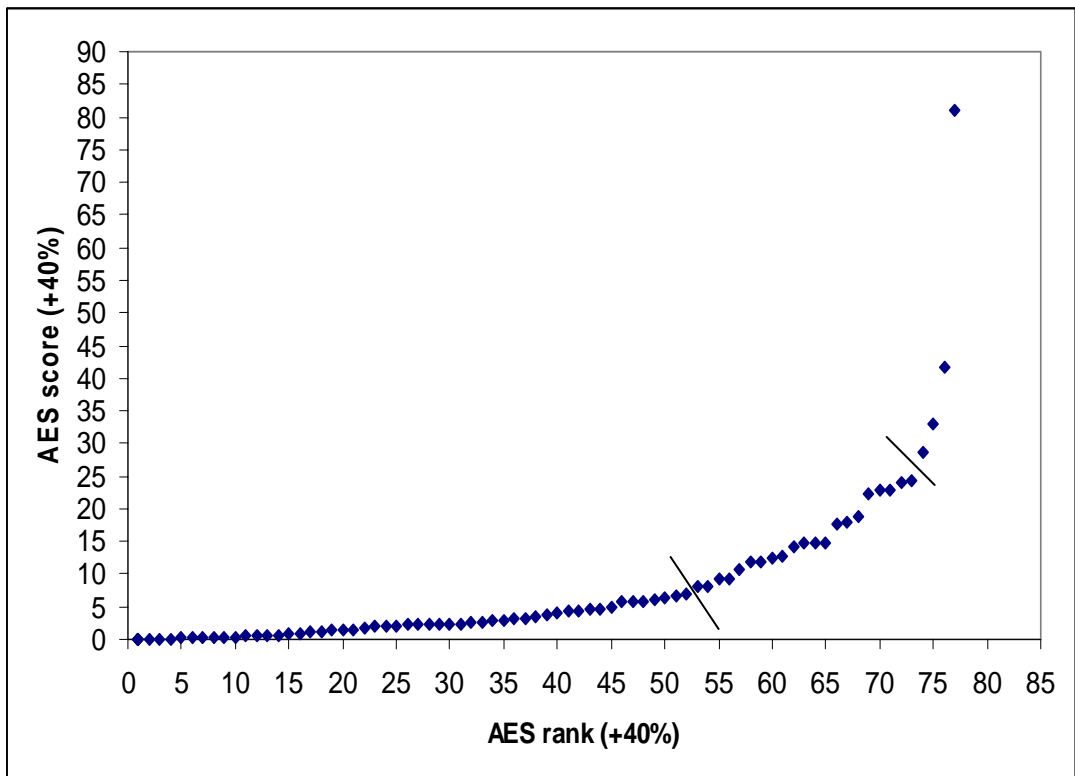


Figure 20. The AES score distribution of the programs for +40% upper bounds

Table 27 shows sorting results of the programs with AES scores by two sets of threshold values applied to the +25% upper bounds case. The threshold column shows when the threshold value is changed and what the new value is. We set the thresholds by trying sets of them on the data. As we do not change them unless we hit the cut-off points, we choose so as to achieve the desired level of density in each range defined by the threshold change points. Also, we set the thresholds in such a way that they result in the opening of a new class whenever we reach a cut-off point. The programs in the classes are listed in order of their AES ranks. Table 28 shows the results of the +40% case by the same threshold sets. If a program is left alone in a class even with adapted thresholds and the DM does not wish to increase the thresholds further, we can conclude that this DMU is an outlier and it should be considered separately.

Let us now sort the programs by their Cross-Efficiency scores presented in the previous section. We will also look at the distribution of Cross-Efficiency scores to see if there are cut-off points in the slopes. Figures 21 and 22 illustrate the Cross-Efficiency score distributions (again in rank order) of the +25% and +40% upper bounds cases respectively. We observe in the graphs that there is a single cut-off point in each case (again shown by lines). For the +25% case, it is right after Program 11, and for the +40% case it is after Program 10. Table 29 shows the sorting results of the programs for both cases with Cross-Efficiency scores and a single set of threshold values. Programs are listed in rank order of their Cross-Efficiency scores.

When we compare the sorting results by the AES scores and Cross-Efficiency scores, we see that they are not alike even though the numbers of classes are similar. The basic reason is the differences between the ranking lists. Another reason is the two methods' score characteristics. AES scores are more suited to discriminate between programs down the ranking list rather than the programs ranking high. However, Cross-Efficiency scores can differentiate between programs that rank high or low while it treats the average performers more equally.

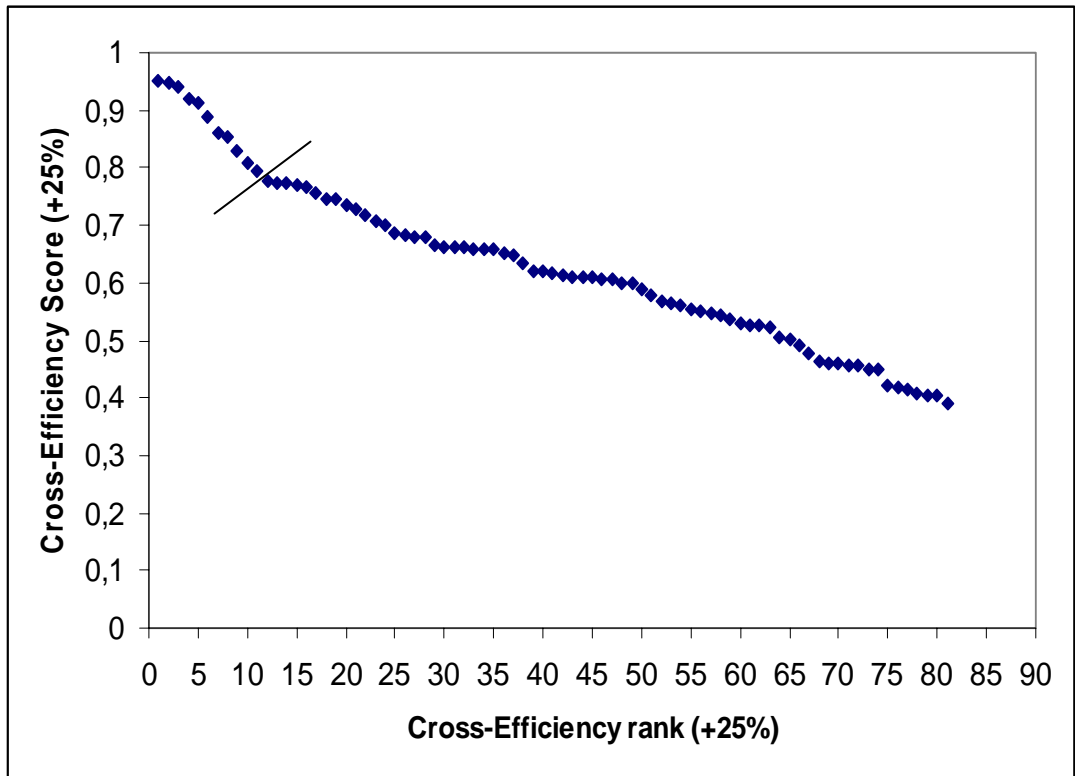


**Table 27. AES sorting results of the +25% upper bounds case**

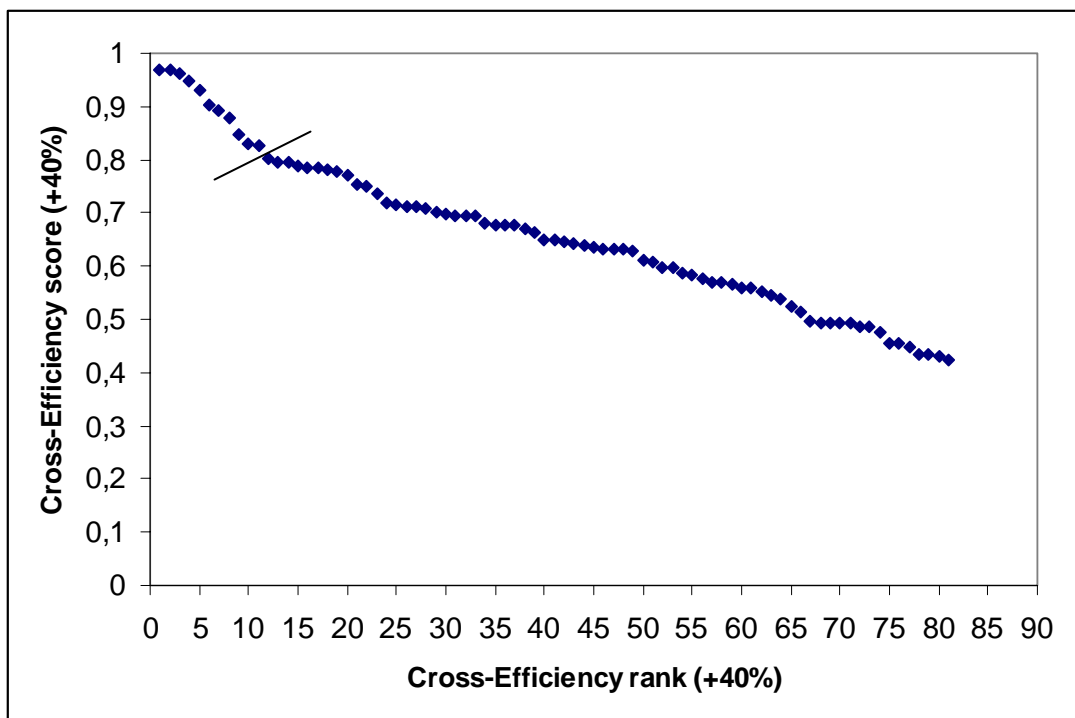
Threshold	Class	Programs	
1	1	1,2,3,5,4,6,7,9,8,10,13,14,11,15,16,12,20,18,19	
	2	21,23,28,24,25,17,22,27,31,32,37,38,26,29	
	3	48,40,45	
	4	46,51,55,47,36,56,54,44	
	5	58,53,52,49,39	
	3	6	59,61,60,64,50,63,66,62,69
		7	65,72
	5	8	73,78,70,80,76
		9	68,75,77
		10	67
		11	79,74
		12	71,57,81
Threshold	Class	Programs	
1	1	1,2,3,5,4,6,7,9,8,10,13,14,11,15,16,12,20,18,19	
	2	21,23,28,24,25,17,22,27,31,32,37,38,26,29	
	3	33,42,30,35,43,34,41	
	4	48,40,45	
	5	46,51,55,47,36,56,54,44	
	6	58,53,52,49,39	
	4	59,61,60,64,50,63,66,62,69,65,72	
	7	73,78,70,80,76,68,75,77	
	9	67,79,74	
	10	71,57,81	

**Table 28. AES sorting results of the +40% upper bounds case**

Threshold	Class	Programs	
1	1	1,2,3,5,7,4,6,9,19,8,11,10,17,13,15,12,18,14,16,20,21,26	
	2	23,24,28,25,35,33,22,27,39,32,30,37,34,29,31,38,42,41,45	
	3	43,40,36,47,48,44,46,51	
	4	55,49,53,56,54,52,58	
	3	5	68,59,61,67,60,64,63,50,66
		6	62,69,72,65
		7	79,76,70
		8	73,78,75,80,71
	5	9	77,57
		10	74
		11	81
Threshold	Class	Programs	
1	1	1,2,3,5,7,4,6,9,19,8,11,10,17,13,15,12,18,14,16,20,21,26	
	2	23,24,28,25,35,33,22,27,39,32,30,37,34,29,31,38,42,41,45	
	3	43,40,36,47,48,44,46,51	
	4	55,49,53,56,54,52,58	
	4	5	68,59,61,67,60,64,63,50,66,62,69,72,65
		6	79,76,70
		7	73,78,75,80,71
	7	8	77,57
		9	74
		10	81



**Figure 21. Cross-Efficiency score distribution of the programs for +25% upper bounds**



**Figure 22. Cross-Efficiency score distribution of the programs for +40% upper bounds**

**Table 29. Cross-Efficiency sorting results of +25% and +40% upper bounds cases**

+25% Upper Bounds			
Threshold	Class	Programs	
0.05	1	1,2,3,5,4,6	
	2	7,9,8,10,11	
0.03	3	14,15,13,12,16,17,18,19,21	
	4	20,24,23,22,28,26,25,30	
	5	37,27,29,33,31,35,34,32,38,36	
	6	43,39,45,42,41,47,49,40,46,44,53,48	
	7	51,50,52,54,55,57,56,58,59,60,62,67,61	
	8	65,63,68,64	
	9	66,69,71,70,74,77,72	
	10	73,79,76,78,80,75,81	
	+40% Upper Bounds		
	Threshold	Class	Programs
0.05	1	1,2,3,5,4	
	2	5,7,9,8,11,10	
0.03	3	14,13,15,19,12,16,18,17,21	
	4	24,20,23,30,22,28,26,25,34,29,33,35,37	
	5	32,27,31,38,36,39,45,43,47,42,49,41,40,46,44,53	
	6	51,48,50,52,54,57,55,56,58,62,59,60	
	7	61,67,65,63,68	
	8	64,77,74,71,70,69,66,72	
	9	73,79,76,75,80,78,81	

## CHAPTER 6

### CONCLUSIONS AND FURTHER RESEARCH AREAS

In this study we address the problem of ranking multi-criteria alternatives. We propose a DEA-based approach, the AES Method, to ranking. Unlike the classical DEA scores and the  $D_k$  values that measure the number of alternatives to leave the set for the efficiency of others, the AES Method is not fully dependent on the relative positioning of the alternatives with respect to each other. Since it considers the change in the efficiency scores of the alternatives while we delete  $0, \dots, D_k$  number of alternatives from the set, the method avoids the negative effect of outliers and crowding in certain areas on some alternatives. It favors alternatives that manage to improve quickly and also alternatives that maintain high levels of efficiency scores. In view of the fact that ranking lists that are not customized will not be of much help to the DMs, we propose the use of weight restrictions to incorporate preference judgments into the analysis.

We apply the AES Method on global MBA programs. We use data extracted from the 2005 Financial Times rankings. We rank 81 programs with 20 criteria according to their AES scores using virtual weight restrictions. We also rank the programs with the Cross-Efficiency Method to enable comparison. We again use virtual weight restrictions in the Cross-Efficiency Method.

We propose the use of an incremental clustering algorithm on the AES scores for sorting the alternatives. We change the threshold values of the sorting algorithm according to the change in the slope of the AES scores of the programs. We also use the Cross-Efficiency scores for sorting the programs.

As the area values of the efficient alternatives are all 0 in the AES Method, we can only discriminate between inefficient ones. Although the use of weight restrictions reduces the number of alternatives with areas of 0, still a number of alternatives may share the first rank. A supplementary approach can be developed for discriminating between the alternatives with 0 areas. The AES Method studies

the improvement in the efficiency scores of the inefficient alternatives while reducing the size of the alternative set. For the efficient alternatives, the improvement in the efficiency scores of others for which they are to be deleted may be studied. If the removal of an efficient alternative brings substantial changes to the efficiency scores of others whereas the removal of another causes insignificant changes, the former alternative may be considered preferable. A new measure can be developed for efficient alternatives after the application of the AES Method.

In this study, the AES Method is applied to the MBA ranking problem which consists of alternatives with only outputs. The method can be applied to problems with multiple inputs and outputs, in this case, AR-II type of restrictions can be employed. Different weight restriction approaches can be applied to study the effects.

## REFERENCES

- Adler, N., Friedman, L., Sinuany-Stern, Z., (2002), "Review of Ranking Methods in the Data Envelopment Analysis Context", *European Journal of Operational Research*, 140, 249-265.
- Ali, A.I., Cook, W.D., Seiford, L.M., (1991), "Strict vs. Weak Ordinal Relations for Multipliers in Data Envelopment Analysis", *Management Science*, 37 (6), 733-738.
- Allen, R., Athanassopoulos, A., Dyson, R.G., Thanassoulis, E., (1997), "Weights Restrictions and Value Judgements in Data Envelopment Analysis: Evolution, Development and Future Directions", *Annals of Operations Research*, 73, 13-34.
- Andersen, P., Petersen, N.C., (1993), "A Procedure for Ranking Efficient Units in Data Envelopment Analysis", *Management Science*, 39 (10), 1261-1264.
- Business Week Online, (2006), "Business Week Online / 2004 B-School Profiles and Rankings", <http://www.businessweek.com/bschools/04/index.html>, last date accessed: 26.07.2006.
- Büyükbaşaran, T., (2005), "Ranking Units by Target-Direction-Set Value Efficiency Analysis and Mixed Integer Programming", MS Thesis, Middle East Technical University.
- Charnes, A., Cooper, W.W., Rhodes, E., (1978), "Measuring Efficiency of Decision Making Units", *European Journal of Operational Research*, 2, 429-444.
- Cook, W.D., (2006), "Distance-Based and Ad Hoc Consensus Models in Ordinal Preference Ranking", *European Journal of Operational Research*, 172, 369-385.
- Cook, W.D., Kress, M., (1990), "A Data Envelopment Model for Aggregating Preference Rankings", *Management Science*, 36(11), 1302-1310.
- Cooper, W.W., Seiford, L.M., Tone, K., (2000), "Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References, and DEA-Solver Software", Kluwer Academic Publishers, USA.
- Doyle, J., Green, R., (1994), "Efficiency and Cross-Efficiency in DEA: Derivations, Meanings and Uses", *The Journal of the Operational Research Society*, 45 (5), 567-578.
- Dyson, R.G., Thanassoulis, E., (1988), "Reducing Weight Flexibility in Data Envelopment Analysis", *The Journal of the Operational Research Society*, 39 (6), 563-576.

Falk, J.E., Karlov, V.E., (2001), "Robust Separation of Finite Sets via Quadratics", *Computers and Operations Research*, 28, 537-561.

Financial Times, (2006), "FT.com / Business Education / Global MBA Rankings 2006", <http://rankings.ft.com/rankings/mba/rankings.html>, last date accessed: 26.07.2006.

Foroughi, A.A., Tamiz, M., (2005), "An Effective Total Ranking Model for a Ranked Voting System", *The International Journal of Management Science*, 33, 491-496.

Golany, B., (1988), "A Note on Including Ordinal Relations among Multipliers in Data Envelopment Analysis", *Management Science*, 34 (8), 1029-1033.

Halme, M., Joro, T., Korhonen, P., Salo, S., Wallenius, J., (1999), "A Value Efficiency Approach to Incorporating Preference Information in Data Envelopment Analysis", *Management Science*, 45 (1), 103-115.

Halme, M., Korhonen, P., (2000), "Restricting Weights in Value Efficiency Analysis", *European Journal of Operational Research*, 126, 175-188.

Jain, A.K., Murty, M.N., Flynn, P.J., (1999), "Data Clustering: A Review", *ACM Computing Surveys*, 31 (3), 264-323.

Köksalan, M., Büyükbaşaran, T., Özpeynirci, Ö., Wallenius, J., (2006), "An Approach to Ranking with an Application to MBA Programs", Technical Report, METU IE Department, 2006-1.

Landsdowne, Z.F., (1996), "Ordinal Ranking Methods for Multicriterion Decision Making", *Naval Research Logistics*, 43, 613-627.

Obata, T., Ishii, H., (2003), "A Method for Discriminating Efficient Candidates with Ranked Voting Data", *European Journal of Operational Research*, 151, 233-237.

Podinovski, V.V., (2004), "Production Trade-Offs and Weight Restrictions in Data Envelopment Analysis", *The Journal of the Operational Research Society*, 55, 1311-1322.

Rousseau, J.J., Semple, J.H., (1995), "Radii of Classification Preservation in Data Envelopment Analysis: A Case Study of 'Program Follow-Through'", *The Journal of the Operational Research Society*, 46 (8), 943-957.

Sarkis, J., (2000), "A Comparative Analysis of DEA as a Discrete Alternative Multiple Criteria Decision Tool", *European Journal of Operational Research*, 123, 543-557.

Sarrico, C.S., Dyson, R.G., (2004), "Restricting Virtual Weights in Data Envelopment Analysis", *European Journal of Operational Research*, 159, 17-34.

Seiford, L.M., (1996), "Data Envelopment Analysis: The Evolution of the State of the Art (1978-1995)", *The Journal of Productivity Analysis*, 7, 99-137.

Seiford, L.M., (1997), "A Bibliography for Data Envelopment Analysis (1978-1996)", *Annals of Operations Research*, 73, 393-438.

Seiford, L.M., Thrall, R.M., (1990), "Recent Developments in DEA: The Mathematical Programming Approach to Frontier Analysis", *Journal of Econometrics*, 47, 7-38.

Sinuany-Stern Z., Mehrez, A., Barboy, A., (1994), "Academic Departments Efficiency via DEA", *Computers and Operations Research*, 21 (5), 543-556.

The, A.N., Mousseau, V., (2002), "Using Assignment Examples to Infer Category Limits for the ELECTRE TRI Method", *Journal of Multi-Criteria Decision Analysis*, 11, 29-43.

Thompson, R.G., Langemeier, L.N., Lee, C.-T., Lee, E., Thrall, R.M., (1990), "The Role of Multiplier Bounds in Efficiency Analysis with Applications to Kansas Farming", *Journal of Econometrics*, 46, 93-108.

US News and World Report, (2006), "USNews.com / America's Best Graduate Schools 2007 / Business / Executive MBA",  
[http://www.usnews.com/usnews/edu/grad/rankings/mba/brief/mbasp07\\_brief.php](http://www.usnews.com/usnews/edu/grad/rankings/mba/brief/mbasp07_brief.php),  
last date accessed: 26.07.2006.

Wong, Y.-H.B., Beasley, J.E., (1990), "Restricting Weight Flexibility in Data Envelopment Analysis", *The Journal of the Operational Research Society*, 41 (9), 829-835.

Zopounidis, C., Doumpos, M., (2002), "Multicriteria Classification and Sorting Methods: A Literature Review", *European Journal of Operational Research*, 138, 229-246.

Zopounidis, C., Doumpos, M., (1999), "A Multicriteria Decision Aid Methodology for Sorting Decision Problem: The Case of Financial Distress", *Computational Economics*, 14, 197-218.



## APPENDIX

**Table A. Rank lists of FT, AES, Cross-Efficiency and their differences**

FT	Ranks				Rank Differences									
	AES +25%	AES +40%	CROSS +25%	CROSS +40%	AES +25-40%	CROSS +25-40%	AES - CROSS +25%	AES - CROSS +40%	AES +25% - FT	AES +40% - FT	CROSS +25% - FT	CROSS +40% - FT		
1	2.5	3	1	1	-0.5	0	1.5	2	1.5	2	0	0		
2	2.5	3	2	2	-0.5	0	0.5	1	0.5	1	0	0		
3	2.5	3	3	3	-0.5	0	-0.5	0	-0.5	0	0	0		
4	5	6	5	5	-1	0	0	1	1	2	1	1		
5	2.5	3	4	4	-0.5	0	-1.5	-1	-2.5	-2	-1	-1		
6	6	7	6	6	-1	0	0	1	0	1	0	0		
7	7	3	7	7	4	0	0	-4	0	-4	0	0		
8	9	10	9	9	-1	0	0	1	1	2	1	1		
9	8	8	8	8	0	0	0	0	-1	-1	-1	-1		
10	10	12	10	11	-2	-1	0	1	0	2	0	1		
11	13	11	11	10	2	1	2	1	2	0	0	-1		
12	16	16	15	16	0	-1	1	0	4	4	3	4		
13	11	14	14	13	-3	1	-3	1	-2	1	1	0		
14	12	18	12	12	-6	0	0	6	-2	4	-2	-2		
15	14	15	13	14	-1	-1	1	1	-1	0	-2	-1		
16	15	19	16	17	-4	-1	-1	2	-1	3	0	1		
17	25	13	17	19	12	-2	8	-6	8	-4	0	2		
18	18	17	18	18	1	0	0	-1	0	-1	0	0		
19	19	9	19	15	10	4	0	-6	0	-10	0	-4		
20	17	20	21	22	-3	-1	-4	-2	-3	0	1	2		

**Table A (continued)**

FT	Ranks				Rank Differences							
	AES +25%	AES +40%	CROSS +25%	CROSS +40%	AES +25-40%	CROSS +25-40%	AES - CROSS +25%	AES - CROSS +40%	AES +25% - FT	AES +40% - FT	CROSS +25% - FT	CROSS +40% - FT
21	20	21	20	20	-1	0	0	1	-1	0	-1	-1
22	26	29	24	25	-3	-1	2	4	4	7	2	3
23	21	23	23	23	-2	0	-2	0	-2	0	0	0
24	23	24	22	21	-1	1	1	3	-1	0	-2	-3
25	24	26	27	28	-2	-1	-3	-2	-1	1	2	3
26	32	22	26	27	10	-1	6	-5	6	-4	0	1
27	27	30	30	35	-3	-5	-3	-5	0	3	3	8
28	22	25	25	26	-3	-1	-3	-1	-6	-3	-3	-2
29	33	36	31	30	-3	1	2	6	4	7	2	1
30	36	33	28	24	3	4	8	9	6	3	-2	-6
31	28	37	33	36	-9	-3	-5	1	-3	6	2	5
32	29	32	36	34	-3	2	-7	-2	-3	0	4	2
33	34	28	32	31	6	1	2	-3	1	-5	-1	-2
34	39	35	35	29	4	6	4	6	5	1	1	-5
35	37	27	34	32	10	2	3	-5	2	-8	-1	-3
36	48	44	38	38	4	0	10	6	12	8	2	2
37	30	34	29	33	-4	-4	1	1	-7	-3	-8	-4
38	31	38	37	37	-7	0	-6	1	-7	0	-1	-1
39	56	31	40	39	25	1	16	-8	17	-8	1	0
40	42	43	46	46	-1	0	-4	-3	2	3	6	6

**Table A (continued)**

FT	Ranks				Rank Differences							
	AES +25%	AES +40%	CROSS +25%	CROSS +40%	AES +25-40%	CROSS +25-40%	AES - CROSS +25%	AES - CROSS +40%	AES +25% - FT	AES +40% - FT	CROSS +25% - FT	CROSS +40% - FT
41	40	40	43	45	0	-2	-3	-5	-1	-1	2	4
42	35	39	42	43	-4	-1	-7	-4	-7	-3	0	1
43	38	42	39	41	-4	-2	-1	1	-5	-1	-4	-2
44	51	47	48	48	4	0	3	-1	7	3	4	4
45	43	41	41	40	2	1	2	1	-2	-4	-4	-5
46	44	48	47	47	-4	0	-3	1	-2	2	1	1
47	47	45	44	42	2	2	3	3	0	-2	-3	-5
48	41	46	50	51	-5	-1	-9	-5	-7	-2	2	3
49	55	51	45	44	4	1	10	7	6	2	-4	-5
50	61	64	52	52	-3	0	9	12	11	14	2	2
51	45	49	51	50	-4	1	-6	-1	-6	-2	0	-1
52	54	55	53	53	-1	0	1	2	2	3	1	1
53	53	52	49	49	1	0	4	3	0	-1	-4	-4
54	50	54	54	54	-4	0	-4	0	-4	0	0	0
55	46	50	55	56	-4	-1	-9	-6	-9	-5	0	1
56	49	53	57	57	-4	0	-8	-4	-7	-3	1	1
57	80	79	56	55	1	1	24	24	23	22	-1	-2
58	52	56	58	58	-4	0	-6	-2	-6	-2	0	0
59	57	58	59	60	-1	-1	-2	-2	-2	-1	0	1
60	59	61	60	61	-2	-1	-1	0	-1	1	0	1

**Table A (continued)**

Ranks				Rank Differences								
FT	AES +25%	AES +40%	CROSS +25%	CROSS +40%	AES +25-40%	CROSS +25-40%	AES - CROSS +25%	AES - CROSS +40%	AES +25% - FT	AES +40% - FT	CROSS +25% - FT	CROSS +40% - FT
61	58	59	63	62	-1	1	-5	-3	-3	-2	2	1
62	64	66	61	59	-2	2	3	7	2	4	-1	-3
63	62	63	65	65	-1	0	-3	-2	-1	0	2	2
64	60	62	67	67	-2	0	-7	-5	-4	-2	3	3
65	66	69	64	64	-3	0	2	5	1	4	-1	-1
66	63	65	68	73	-2	-5	-5	-8	-3	-1	2	7
67	76	60	62	63	16	-1	14	-3	9	-7	-5	-4
68	73	57	66	66	16	0	7	-9	5	-11	-2	-2
69	65	67	69	72	-2	-3	-4	-5	-4	-2	0	3
70	70	72	71	71	-2	0	-1	1	0	2	1	1
71	79	77	70	70	2	0	9	7	8	6	-1	-1
72	67	68	74	74	-1	0	-7	-6	-5	-4	2	2
73	68	73	75	75	-5	0	-7	-2	-5	0	2	2
74	78	80	72	69	-2	3	6	11	4	6	-2	-5
75	74	75	80	78	-1	2	-6	-3	-1	0	5	3
76	72	71	77	77	1	0	-5	-6	-4	-5	1	1
77	75	78	73	68	-3	5	2	10	-2	1	-4	-9
78	69	74	78	80	-5	-2	-9	-6	-9	-4	0	2
79	77	70	76	76	7	0	1	-6	-2	-9	-3	-3
80	71	76	79	79	-5	0	-8	-3	-9	-4	-1	-1
81	81	81	81	81	0	0	0	0	0	0	0	0