

MULTIPLIER ADJUSTMENT IN DATA ENVELOPMENT ANALYSIS

Jorge Azevedo Santos (Universidade de Évora)
Luís Cavique Santos (Universidade Aberta)
Armando Brito Mendes (Universidade dos Açores).

Abstract – Weights restriction is a well-known technique in the DEA field. When those techniques are applied, weights cluster around its new limits, making its evaluation dependent of its levels. This paper introduces a new approach to weights adjustment by Goal Programming techniques, avoiding the imposition of hard restrictions that can even lead to unfeasibility. This method results in models that are more flexible.

Keywords: *Data Envelopment Analysis, Efficiency, Weights Restriction, Evaluation, Goal Programming.*

I. INTRODUCTION

Data Envelopment Analysis (DEA) is a mathematical programming based technique to evaluate the relative performance of organisations. While the main applications have been in the evaluation of not-for-profit organisations, the technique can be successfully applied to other organisations, as a recent evaluation of banks in India has demonstrated [1].

With this paper, we have two objectives in mind. The first one is to present DEA-Data Envelopment Analysis, a technique which may have useful applications in many evaluation contexts, namely when assessing not-for-profit organisations. In addition to allowing the ranking of the organisations traditionally termed decision-making units, DEA also creates the conditions to improve performance through target setting and role-model identification. We also briefly describe the technique of deleted domain, also known as Superefficiency.

The second objective is to introduce an entirely new way of adjusting multipliers by means of Goal Programming Techniques. This adjustment is a much more general way of dealing with the incorporation of exogenous structure preferences so far relying only in weights restriction techniques, which, in our point of view leads to the concentration of the weights in its upper and lower limits.

DEA is suited for this type of evaluation because it enables results to be compared making allowances for factors [2]. DEA makes it possible to identify efficient and inefficient units in a framework where results are considered in their particular context. In addition, DEA also provides information that enables the comparison of each inefficient unit with its "peer group", that is to say,

a group of efficient units that are identical with the units under analysis. These role-model units can then be studied in order to identify the success factors that other comparable units can attempt to follow. Thanassoulis et al [3] argue that DEA is preferable to other methods, such as regression analysis, which also make it possible to contextualize results.

The present paper is structured as follows. The next section describes the development and fields of application of the technique, while section III introduces the DEA models followed by a numerical example. In section IV, we present Superefficiency evaluation, an extension of DEA also known as deleted domain. Section V and VI deal with the graphical solution in the weights space and makes a very short description of the weights restrictions technique respectively.

In section VII, a new concept of multiplier adjustment is introduced and exemplified through a small data set.

In section VIII, a case with artificially generated data is solved to highlight the potentialities of this technique. This paper ends up with a final section with the conclusions and directions of future work. Readership not familiar with DEA, may find the brief introduction to the method presented below useful, but for those who wish to follow the matter further there is a good review of DEA in Boussofiane et al [4].

II. HISTORY AND APPLICATIONS OF DEA

DEA is a mathematical programming technique presented in 1978 by Charnes, Cooper and Rhodes [5], although its roots may be found as early as 1957 in Farrell's seminal work [6]. This technique is usually introduced as a non-parametric one, but in fact, it rests on the assumption of linearity [7] and for the original models even in the more stringent assumption of proportionality.

Its application has been focused mainly on the efficiency assessment of not-for-profit organizations, since these cannot be evaluated on the basis of traditional economic and financial indicators used for commercial companies.

The first application of DEA was in the field of Education, in the analysis of the *Program Follow Through*, conducted in the USA, in the late seventies [8]. Since then, it has been used to assess efficiency in areas such as health [9, 10], prisons [11], courts [12], universities and many other not-for-profit sectors. Nowadays, DEA can be seen to have spread to other fields such as Transit [13], Mining [14], Air Transportation [15], and even Banking and Finance [16].

However, many applications belong to the field of education and range from primary education [17, 18], to secondary [19, 20, 21] and university levels [22].

In Data Envelopment Analysis, the organizational units to be assessed should be relatively homogeneous and were originally termed Decision Making Units. As the whole technique is based on, the comparison of each DMU with all the remaining ones a considerable large set of units is necessary for the assessment to be meaningful. We will assume that each DMU produces N outputs by means of M inputs.

III. DEA FORMULATIONS WITH A NUMERICAL EXAMPLE

In DEA, efficiency (h_j) of a specific decision making unit (DMU j) is defined as the ratio between a weighted sum of its N outputs Y_{nj} and a weighted sum of its M inputs X_{mj} , a natural extension of the concept of efficiency used in the fields of physics and engineering [23]:

$$h_j = \frac{\sum_{n=1}^N v_{nj} y_{nj}}{\sum_{m=1}^M \mu_{mj} x_{mj}} \quad (1)$$

When assessing a set of J organisations, where X_{mj} stands for the m^{th} input of the j^{th} DMU, with a similar meaning for Y_{nj} , the weights μ_{mj} and v_{nj} , in expression (1), are chosen for each j DMU under evaluation as those that maximize its efficiency as defined by h_j . Several constraints have to be added to the maximization problem:

- The strict positivity [24] of the weights μ_{mj} , v_{nj} (also known as virtual multipliers or simply as “multipliers”).
- For scaling purposes, all J DMUs under analysis must have efficiencies not exceeding an agreed value, typically one or 100%, as is usual in engineering definitions of efficiency.
- A third kind of restriction has to be included, since otherwise this linear fractional program would yield an infinite number of solutions. In fact, if a set of weights μ_{mj} , v_{nj} returns the optimal solution, so would $k\mu_{mj}$, kv_{nj} . Making the denominator, in Express-

sion (1), equal to one or 100%, circumvents this situation.

Therefore, we have to solve the following maximization problem **for each one** of the J DMUs under analysis:

$$\text{Max } h_j = \frac{\sum_{n=1}^N v_{nj} y_{nj}}{\sum_{m=1}^M \mu_{mj} x_{mj}} \quad (2)$$

$$\text{s.t. } \mu_{mj} > 0 \quad m=1\dots M \quad (3)$$

$$v_{nj} > 0 \quad n=1\dots N \quad (4)$$

$$h_j = \frac{\sum_{n=1}^N v_{nj} y_{nj}}{\sum_{m=1}^M \mu_{mj} x_{mj}} \leq 1 \quad j=1\dots J \quad (5)$$

$$\sum_{m=1}^M \mu_{mj} x_{mj} = 1 \quad (6)$$

This Fractional Linear Program can be solved by means of the Charnes and Cooper transformation [25] which yield the following Linear Program:

$$\text{Max } h_j = \sum_{n=1}^N v_{nj} y_{nj} \quad (7)$$

$$\text{s.t. } \sum_{m=1}^M \mu_{mj} x_{mj} = 1 \quad (8)$$

$$\sum_{n=1}^N v_{nj} y_{nj} \leq \sum_{m=1}^M \mu_{mj} x_{mj} \quad j=1\dots J \quad (9)$$

$$\mu_{mj} \geq \varepsilon > 0 \quad m=1\dots M \quad (10)$$

$$v_{nj} \geq \varepsilon > 0 \quad n=1\dots N \quad (11)$$

The problem above is known as the multiplier problem, since its unknowns are the weights, which are usually lower bounded by a small quantity ε (Expressions: 10-11) so that all Inputs and Outputs are considered in the evaluation [24], even if with a minor weight ε , set in all the following formulations equal to 10^{-6} .

The dual of this problem, which we shall call the envelopment problem, provides important information about economies that could be achieved in all the inputs; it also indicates which efficient units the inefficient unit being assessed should emulate. Those efficient units are usually referred to as the reference set or peer group of the unit under evaluation.

To illustrate the Data Envelopment Analysis technique, an example is introduced in Table I, with 12 DMUs producing two Outputs Y_1 and Y_2 from a single Input X_1 , under the assumption of constant returns to

scale, which simply means that if one doubles the Inputs of any unit it would be expected that its Outputs would also double. In algebraic form, this can be stated as: if x_j yields Outputs y_j then Inputs kx_j should produce Outputs ky_j .

Table I- Outputs normalised by Input X1

DMU	x1	y1	y2	y1/x1	y2/x1
1	4.0	2.0	28.0	.500	7.000
2	5.0	1.0	22.5	.200	4.500
3	6.0	6.0	12.0	1.000	2.000
4	10.0	8.0	60.0	.800	6.000
5	11.0	7.0	16.5	.636	1.500
6	8.0	6.0	12.0	.750	1.500
7	9.0	7.0	6.0	.778	.667
8	5.0	3.0	30.0	.600	6.000
9	5.5	4.4	5.5	.800	1.000
10	8.0	4.0	72.0	.500	9.000
11	10.0	2.0	20.0	.200	2.000
12	8.0	1.0	4.0	.125	.500

In this simple example, we can normalise the Outputs by the only Input and plot them in the plane. This is also equivalent to consider that we are dealing with a constant input of 1. This way we will be working in the plane defined by $x=1$ in the 3 dimensional one input two outputs space.

From Figure 1 it is easy to understand the reason for naming this technique Data Envelopment Analysis; in fact each DMU is analysed against the envelope of the most efficient units. For instance, the efficiency of DMU 8 is 0.857 (see Table II). This means that it could reduce its input to 85.7% of its current value reaching its target, $C_i 8$ (where C_i stands for Composite unit under minimisation of inputs), which is the same as for DMU 11 except that, for the latter, a reduction to 28.6% of X_1 's current level of inputs would be necessary for this DMU to become efficient, since its efficiency is only 0.286.

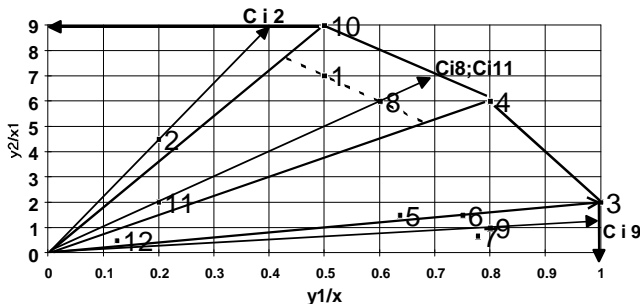


Figure 1- Efficient frontier and radial projections for inefficient units.

Although the results could be obtained graphically, we present in table II the results obtained by any standard linear optimisation software.

Table II- Results for the multiplier problem

DMU	μ_1	v_1	v_2	Efficiency
1	0.250	0.179	0.018	0.857
2	0.200	0.000	0.022	0.500
3	0.167	0.152	0.008	1.000
4	0.100	0.091	0.005	1.000
5	0.091	0.083	0.004	0.647
6	0.125	0.114	0.006	0.750
7	0.111	0.111	0.000	0.778
8	0.200	0.143	0.014	0.857
9	0.182	0.182	0.000	0.800
10	0.125	0.089	0.009	1.000
11	0.100	0.071	0.007	0.286
12	0.125	0.114	0.006	0.136

IV. SUPEREFFICIENCY / DELETED DOMAIN EXTENSION

We arrive at the concept of Superefficiency by allowing the efficiency of the DMU being assessed to be greater than unity. This is achieved by removing the corresponding constraint from the set of J constraints in Expression (9). This is the reason why this technique is also known as deleted domain. The Superefficiency only affects units deemed as efficient, as the removed constraint is not binding for the inefficient units, since their efficiency is, by definition, less than unity.

This extension to DEA was first published by Andersen and Petersen [26] and its use is strongly recommended by the authors as a consequence of its simplicity and usefulness.

By using Superefficiency, it is possible to rank all units, even the efficient ones that by standard DEA techniques would all be rated as equal - their efficiency having reached the top value of 100%.

For the example presented in the previous section, the Superefficiency for the 3 efficient DMUs would be as presented in table III.

Table III- Superefficiency scores for the efficient units

Unit	Superefficiency
Unit 4	107.50%
Unit 3	125.00%
Unit 10	128.57%

Units 3 and 10 are efficient and robust, while any small increase in the Input or decrease in the Outputs of Unit 4 may make it inefficient.

An important additional benefit from this extension to the DEA model is that the set of weights is uniquely determined for the efficient units in all practical applications [21].

V. GRAPHICAL SOLUTION

The linear program defined by expressions (7) through (11) can be solved by the traditional graphical method if we have to deal with only 2 variables.

To reduce the problem from 3 to 2 variables, we will exploit the fact that we assume to be working under the constant returns to scale assumption and so we will scale all data to unity input level, so that our new data set will be the following one:

DMU	Xn1	Yn1=y1/x1	Yn2=y2/x1
1	1	0.500	7.000
2	1	0.200	4.500
3	1	1.000	2.000
4	1	0.800	6.000
5	1	0.636	1.500
6	1	0.750	1.500
7	1	0.778	0.667
8	1	0.600	6.000
9	1	0.800	1.000
10	1	0.500	9.000
11	1	0.200	2.000
12	1	0.125	0.500

Table IV- Normalized data to unity input level

Since $x_{n1}=1$ its multiplier $\mu_{n1}=1$ and we have to solve the linear program just for v_{n1} and v_{n2} , we will illustrate the results from the software EMS developed by Holger Scheel that uses an interior point solver.

Its results are presented in figure 2

DMU	Score	x1 {0}\{w}	y1 {0}\{w}	y2 {0}\{w}	Benchmarks	{S} x1 {0}	{S} y1 {0}	{S} y2 {0}		
1	F1	85.71%	1.00000000	0.71428571	0.07142857	4 (0.24)	10 (0.62)	0.00	0.00	0.00
2	F2	50.00%	1.00000000	0.00000000	0.11111111	10 (0.50)		0.00	0.05	0.00
3	F3	125.00%	1.00000000	1.25000000	0.00000000	5				
4	F4	107.50%	1.00000000	0.87500000	0.06250000	5				
5	F5	64.64%	1.00000000	0.90909091	0.04545454	3 (0.59)	4 (0.05)	0.00	0.00	0.00
6	F6	75.00%	1.00000000	0.99815962	0.00092019	3 (0.75)		0.00	0.00	0.00
7	F7	77.80%	1.00000000	1.00000000	0.00000000	3 (0.78)		0.00	0.00	0.89
8	F8	85.71%	1.00000000	0.71428571	0.07142857	4 (0.57)	10 (0.29)	0.00	0.00	0.00
9	F9	80.00%	1.00000000	1.00000000	0.00000000	3 (0.80)		0.00	0.00	0.60
10	F10	128.57%	1.00000000	0.00000000	0.14285714	4				
11	F11	28.57%	1.00000000	0.71428566	0.07142857	4 (0.19)	10 (0.10)	0.00	0.00	0.00
12	F12	13.64%	1.00000000	0.90909091	0.04545455	3 (0.08)	4 (0.06)	0.00	0.00	0.00

Figure 2- Results obtained with the software EMS.

We can confirm the correctness of our results, namely that $\mu_{n1}=1$ and that the values obtained for the efficiency are equal to those previously presented in table II.

It is worthy to note that DMUs F3, F7 and F9 place a minimal weight on Output2 while DMUs F2 and F10 choose to ignore Output1. This kind of problem is usually solved by a technique known as weight restrictions in the sense of avoiding such a flexible set of weights and incorporating value judgments.

By solving graphically, we get the graph shown in figure 3 where we have 12 line segments, one for each constraint. The feasible region is the intersection of the twelve (lower) half planes containing the origin. It is

clear that inefficient units like F12 correspond to non-binding constraints and that the objective function is parallel to the respective DMU constraint.

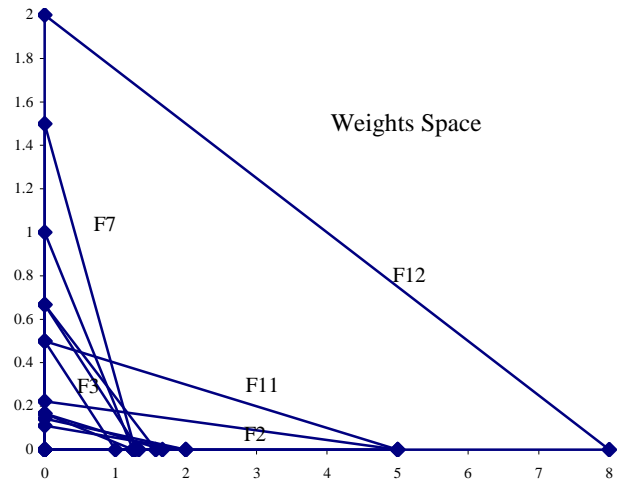


Figure 3- Restrictions imposed by the 12 units in the Weights Space

For the sake of clarity, we will expand the previous picture just to the efficient units, in order to highlight the feasible region defined by the pentagon ABCD0 (in the clockwise direction). This feasible region is always the same for inefficient units for the efficient ones, it depends on the kind of model we are working with.

In the case of the traditional model the feasible region is always identical, in the deleted domain case, also known as Superefficiency, the constraint related to the DMU being analysed is deleted, and even a non efficient Unit can pop-up in its reference set [21], this is the case of unit 10. In fact, its reference set is unit 1, since the feasible region on the weights space is EFCDO (in the clockwise direction) and the objective function is parallel to the line segment 10, where it takes the unity value.

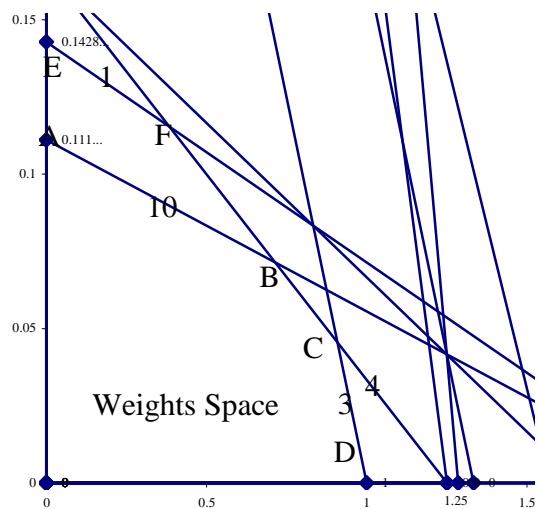


Figure 4- Detail around the origin of the constraints imposed by the 12 units in the Weights Space

VI. WEIGHTS RESTRICTIONS

To avoid a given DMU “to choose” a rather unbalanced set of weights (as is the case of units F2 and F10 which ignore Output 1 while F3, F7 and F9 ignore Output 2), it is current practice to place some restrictions on the weights or in the virtual inputs/outputs.

This is a usual way to incorporate judgment values and increase the discriminating potential of the model.

Restrictions on weights can be divided in two broad categories: Relative and absolute weights restrictions. As far as we know, only linear weights restrictions have been considered in the literature, thus we may present the weights restrictions in matrix form as follows:

$$\mathbf{B}^x \boldsymbol{\mu}^T + \mathbf{B}^y \mathbf{v}^T \leq \mathbf{C}$$

Where $\mathbf{B}^x \in \mathcal{R}^{c \times m}$ and $\mathbf{B}^y \in \mathcal{R}^{c \times n}$.

Dimension c refers to the number of constraints.

If $\mathbf{C} \neq \mathbf{0}$ we refer to them as absolute weights restrictions. The absolute weights restrictions are typically imposing a range for an individual weight. This approach was developed by Dyson and Thanassoulis in 1988 [27] and generalized by Roll et al in 1991 [28]. Virtual weights restrictions introduced by Wong and Beasley in 1990 [29] belong to this category also.

If $\mathbf{C} = \mathbf{0}$ we speak about relative weights restrictions since if μ_0, v_0 is a feasible solution so is $k \mu_0, k v_0$. The class of relative weights restrictions includes, among others, the assurance region models of Type I or II by Thompson et al in 1986 [30] and 1990 [31] respectively, as well as cone ratio DEA models from Charnes et al. in 1989 [32].

We can say that the most general approach is to restrict the weights to belong to a closed set, being it a polytope or a polyhedral cone.

VII. WEIGHT ADJUSTMENT BY GOAL PROGRAMMING

With standard DEA, it is common that many weights are null in the optimal solution. One way to avoid this situation is to place restrictions on the weights, but, in this case, weights typically used to cluster in the upper or lower limits. By including some non-linear but convex penalty in the objective function, penalizing deviations from a preference region in the weights space, it is possible to have a more uniform distribution of the weights.

This can be accomplished by the model, described by expressions 12 through 18 where G stands for Global objective function, $P(\vec{d})$ is a penalizing function of the deviations d_m and d_n of the weights from the exogenously imposed goals g_m and g_n .

The penalizing function will always be a convex one, for avoiding difficulties with local minima; it is worth recalling that since the feasible region is convex and the symmetric of the objective function is also convex, a local maximum is also a global maximum [33]. The definition of efficiency remains unchanged, so that this model just adjusts multipliers by a penalizing function

appended to the objective function scaled by a constant k .

$$\text{Max } G_j = \sum_{n=1}^N v_{nj'} y_{nj'} - k \times P(\vec{d}) \quad (12)$$

$$\text{s.t. } d_m = \mu_{mj'} - g_m \quad m=1 \dots M \quad (13)$$

$$d_n = v_{nj'} - g_n \quad n=1 \dots N \quad (14)$$

$$\sum_{m=1}^M \mu_{mj'} x_{mj'} = 1 \quad (15)$$

$$\sum_{n=1}^N v_{nj'} y_{nj'} \leq \sum_{m=1}^M \mu_{mj'} x_{mj'} \quad j=1 \dots J \quad (16)$$

$$\mu_{mj'} \geq \varepsilon > 0 \quad m=1 \dots M \quad (17)$$

$$v_{nj'} \geq \varepsilon > 0 \quad n=1 \dots N \quad (18)$$

We will illustrate this technique with a simple example originating from the previous data set with 12 DMUs.

The preferred location for the weights is around the line $0.4v_{1j} = 4v_{2j}$. We are not interested in justifying this choice, neither other details like the value for the constant k , or the explicit kind of penalizing function, we also remind that our goal is just to exemplify a way to adjust weights in a smoother way than the usual hard restriction techniques do.

Therefore, the Global objective function will take the following form:

$$G_j = \underbrace{\sum_{n=1}^N v_{nj'} y_{nj'}}_{L=\text{Efficiency}} - \underbrace{2 \times (0.4v_1 - 4v_2)^2}_{\text{Penalty (Strictly Convex)}} \quad (19)$$

$-k \times P(\vec{d}) \xrightarrow{\text{is}} \text{Strictly Concave}$

We will now illustrate the graphical solution for the evaluation of Unit 10 under the constant returns to scale assumption and deleted domain technique.

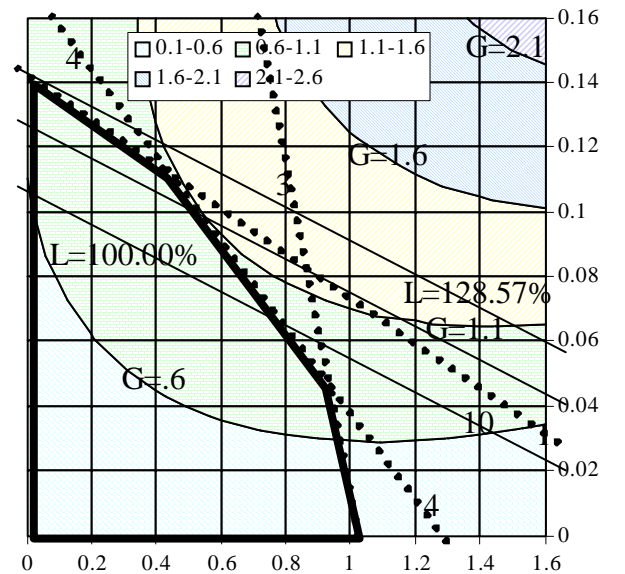


Figure 5- Graphical solution for the evaluation of Unit 10

In figure 5, the dotted lines correspond to constraints related to units 1, 3 and 4. The solid curves represent the isoquants of the Global objective function for $G=0.6$, 1.1, 1.6 and finally 2.1.

We can also see 3 solid straight lines:

1. The isoquant for $L=100.00\%$ that coincides with the constraint that was removed because of the deleted domain technique.
2. The isoquant for $L=114.06\%$ as determined by the exact quadratic programming solution shown in table V.
3. The isoquant for $L=128.57\%$ that equals the score value from the Superefficiency technique depicted in figure 2.

The optimal solution of the Superefficiency CCR model is the basic solution defined by the intersection of the constraint relative to unit 1 and $\mu_1 \geq \varepsilon > 0$. It is interesting to note that an inefficient unit (DMU 1) is defining the optimal solution for an efficient one, a fact first published in 1994 [21] by Santos and Themido.

The solution of the quadratic program can be obtained in a rough manner by the graphical method as exemplified in figure 5, or by the results shown in the line corresponding to DMU 10 in Table V.

Since the constraint relative to DMU 4 is the only binding one its Lagrange multiplier is the only one non-zero ($\lambda_4=1.000$).

The optimum value for the objective function is $G=1.070$, which occurs at the point $v_1=0.51$ $v_2=0.0982$, deviating $d=-0.188$ from the preferred linear relation between weights, as a result we get the final value for efficiency of $L=114.06\%$. This value is lower than the score obtained by the EMS software, since this new result has multipliers that are more desirable in the point of view of the incorporated weights preference structure, by the preferred linear relation between weights.

D	v_1	v_2	G	d	L	λ_3	λ_4	λ_{10}
1	.71	.0714	0.857	0.000	85.71%	.000	0.24	.619
2	.68	.0732	0.465	-0.020	46.59%	.000	0.00	.464
3	.92	.0434	0.934	0.196	101.15%	.000	0.86	.000
4	.87	.0625	1.055	0.100	107.50%	.245	0.00	.790
5	.83	.0547	0.589	0.117	61.60%	.000	0.56	.000
6	.87	.0504	0.686	0.147	72.97%	.000	0.64	.000
7	.90	.0455	0.671	0.182	73.76%	.014	0.59	.000
8	.71	.0714	0.857	0.000	85.71%	.000	0.57	.286
9	.90	.0459	0.707	0.179	77.04%	.000	0.64	.000
10	.51	.0982	1.070	-0.188	114.06%	.000	1.00	.000
11	.71	.0714	0.286	0.000	28.57%	.000	0.19	.095
12	.73	.0692	0.125	0.016	12.60%	.000	0.13	.000

Table V- Exact quadratic programming solutions

The objective function depends on the DMU being evaluated, in the previous picture DMU 10 was considered; now we will show the graphical solution for DMU 3.

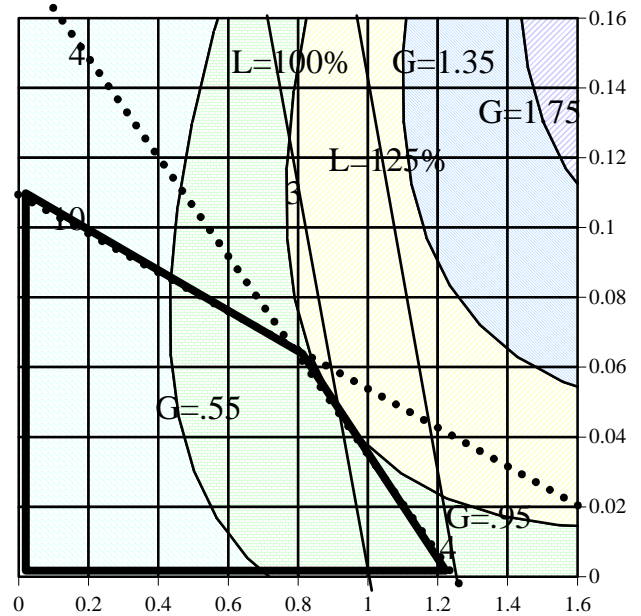


Figure 6- Graphical solution for the evaluation of Unit 10

In figure 6 the dotted lines correspond to constraints related to units 10 and 4. The solid curves represent the isoquants of the Global objective function for $G=0.55$, 0.95, 1.35 and finally 1.75.

We can also see 2 solid straight lines:

1. The isoquant for $L=100.00\%$ that coincides with the constraint that was removed because of the deleted domain technique, very close to the isoquant for $L=101.15\%$ as determined by the exact quadratic programming solution shown in table V.
2. The isoquant for $L=125.00\%$ that equals the score value from the Superefficiency technique depicted in figure 2.

It is interesting to remark that the quadratic objective function isoquants have a different orientation from those depicted in figure 5, since the slope of the linear traditional linear objective functions were different too.

Since the constraint relative to DMU 4 is the only binding one its Lagrange multiplier is the only one non-zero ($\lambda_4=0.857$).

The optimum value for the objective function is $G=0.934$, which occurs at the point $v_1=0.92$ $v_2=0.0434$, deviating $d=0.196$ from the preferred linear relation between weights, as a result we get the final value for efficiency of $L=101.15\%$. Again, this value is lower than the score obtained by the EMS software, since this new result has multipliers that are more evenly distributed, around our goal: the line $v_2=0.1v_1$. The results without weight adjustment were an efficiency score of

$L=125.00\%$ but with a weight pattern neglecting output 2 ($v_1=1.00$ $v_2=0$).

We can compare the score efficiencies obtained by the two models. The results are as depicted in figure 7

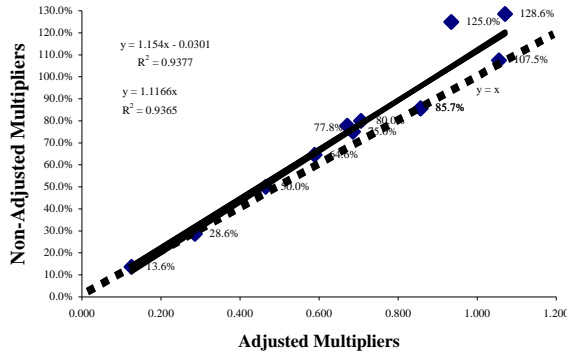


Figure 7- Comparison of efficiency scores

The dotted line is the line defined by $y=x$ (restricted to the first orthant), to make it easy to see that all points are above or on that line, meaning that efficiency scores obtained with weight adjustment are not bigger than the ones obtained without adjustment. There are 3 DMUs (1, 8 and 11) which efficiency score remained unchanged, as a result of the fact that their optimal weights as shown in the EMS results in figure 2 were already complying with our later specification of staying on the line $v_2=0.1v_1$.

We can get the full detailed results for the other remaining 10 DMUs by table V and locating their optimal weights in figures 5 or 6.

It is now worth noting that DMUs 1 and 8 have exactly the same score and the same optimal weights, since the relation $v_2=0.1v_1$ implies an intrinsic marginal rate of substitution of y_1 by y_2 of $v_1/v_2=10$ which is exactly the case for their outputs. The fact that two of its Lagrange multipliers are the only non-zero ones (λ_4 and λ_{10}) means that its optimal weights are in the intersection of the two straight lines relative to the binding restrictions of DMUs 4 and 10.

Now we will introduce a last graphical example of the solution of the CCR Model of the data set presented in figure 8 where we consider one input $X=1$ and two outputs YC_1 and YC_2 whose name comes from the appearance of the 99 points in circular layers in the first orthant.

It is time to explain that the solver for non-linear programs we are using is limited to 100 constraints and this is the reason why we do not solve bigger instances.

In the figure 9, we also present the results of the EMS software under deleted domain and Constant Returns to Scale assumption.

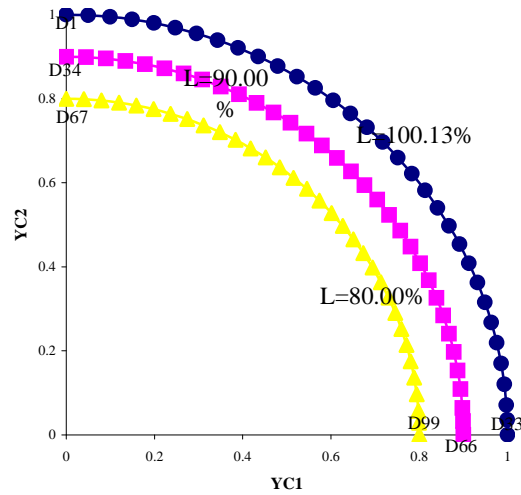


Figure 8- Data for an example of absolute weights adjustment

This time the deviation at the penalizing function was defined as the squared distance from the average of the standard DEA weights normalized by its standard deviation.

$$P(\vec{d}) = \left(\frac{(v_1 - \mu_{v_1})}{\sigma_{v_1}} \right)^2 + \left(\frac{(v_2 - \mu_{v_2})}{\sigma_{v_2}} \right)^2 \quad (20)$$

The meanings of the parameters in expression 20 are as follows:

v_j is the unknown weight for YC_j .

μ_{v_j} is the average of the weight v_j determined solving the standard DEA CCR model under deleted domain.

σ_{v_j} is the standard deviation of the weight v_j found solving the standard DEA CCR model under deleted domain.

The efficiency scores for this absolute adjustment are shown in figure 9, where the highest efficiency score of 100.13% is attained by a single DMU. In fact, the 17th DMU has the same weights than those obtained by the standard minimization of inputs DEA CCR model under deleted domain

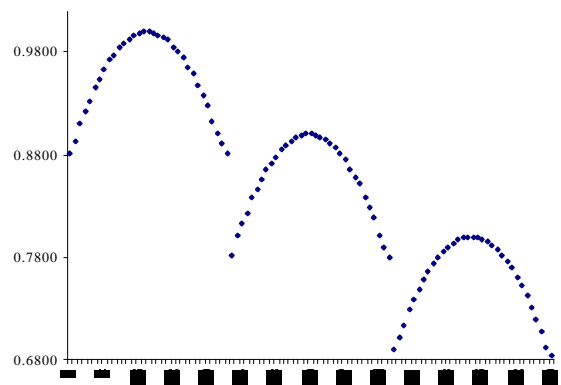


Figure 9- Efficiency scores for absolute adjustment

In figure 10, we present the resulting weights that instead of clustering at the upper and lower limits of its absolute weights restrictions, as happens with traditional weights restriction techniques, now they spread around the central previously defined value $v=(0.6;0.6)$.

We do not know of any other public work that can accomplish such versatility as this technique.

This method can be further enhanced by the introduction of special types of penalty functions like Tchebychev polynomials, or maximally flat polynomials.

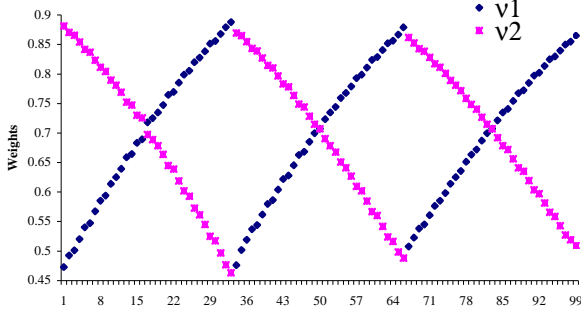


Figure 10- Optimal weights distributing around 0.6

VIII. APPLICATION TO A CASE WITH ARTIFICIALLY GENERATED DATA

Finally we will apply this multiplier adjustment technique to a case with two outputs and four inputs, with simulated data.

Data generating process

When generating the data for the experiments reported in this section we considered two factors: production technology, and inefficiency distribution

Production technology

We considered a two outputs and four inputs production technology. This was obtained through the generation of a single aggregate output incorporating an inefficiency stochastic factor and later split in two outputs as suggested in 1997 by Durchholz and Barr [34].

The single Cobb-Douglas aggregate output and four inputs production technology is specified in terms of its efficient production function $z = f(x_1, x_2, x_3, x_4)$, where z represents the maximum aggregate output that can be produced from the levels x_1, x_2, x_3 and x_4 of the four inputs. Specifically, we used the following shifted Cobb-Douglas production function:

$$z = a_0 \prod_{i=1}^4 (x_i - \alpha_i)^{\beta_i} \quad (21)$$

a_0 is a constant scale factor.

α_i is the shift from the origin for input i .

β_i is the factor elasticity for input i .

z is the single Cobb-Douglas aggregate output

If $\sum \beta_i = 1$, then only constant returns to scale exist in the production process. For $\sum \beta_i < 1$, decreasing returns to scale are present, while $\sum \beta_i > 1$ indicates increasing returns to scale.

In DEA, increasing returns to scale are not used because the function results in only a few DEA efficient points, since for higher levels of input than those defined by the Most Productive Scale Size the Production Possibilities Set is the polyhedral cone from the CCR model. Although this does not pose a problem, it does not represent realistic data.

We made $a_0 = 1$, $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 5$, the input levels x_1, x_2, x_3 and x_4 were generated randomly from four independent uniform probability distributions over the interval $[10, 20]$, and the coefficients $\beta_1, \beta_2, \beta_3$, and β_4 were randomly generated from independent uniform probability distributions over the interval $[0.20, 0.25]$. Since the sum of $\beta_1, \beta_2, \beta_3$, and β_4 is less than one, the production function in expression 21 satisfies the DEA-BCC models assumption of a strictly concave production function, while the shifts $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 5 > 0$ allow both increasing and decreasing returns to scale to prevail.

Inefficiency distribution

We generated the logarithm of the inefficiency,

$u = \ln \theta$ from a half-normal distribution $|N(0; \sigma_u^2)|$,

where the parameter σ_u^2 itself is drawn from a uniform distribution on the interval $[0, 0.1989]$. The range of values for the distribution of σ_u^2 is chosen in such a way that the mean efficiency given by

$$E(h=1/\theta) = e^{-\sigma_u^2 \sqrt{2/\pi}} \quad \text{is between 0.7 and 1.0.}$$

Simulated observations

We first generated random values for β_i between 0.2

and 0.25, and a value σ_u^2 between 0 and 0.1989. Next, we simulated 90 observations of the four inputs x_1 through x_4 between 10 and 20. Those values were then substituted into the efficient production function specified in expression 21 to obtain the corresponding values $z_j = f(x_{ij})$ for the efficient output quantity. Then, we randomly generated the logarithm of actual inefficiency values $u_k = \ln \theta_k$, for each one of the 90 observations from

the half-normal distribution $|N(0; \sigma_u^2)|$.

Finally, we obtained the values for observed aggregated output quantities y_j as: $y_j = f(x_{ij})/\exp(u_j)$ and the true efficiency value $h_k = 1/\exp(u_k)$.

Once the single aggregate output level y was calculated, the two individual output levels were determined by assigning each individual output as a percentage of the aggregate. The percentages for each individual output were drawn from normal distributions with predetermined means and stan-

dard deviations. The means of the normal distributions were chosen so that the percentages sum up to one. In our specific case, we chose each normal distribution to have a mean of 0.5 and a standard deviation of 0.1.

The results from the unbounded model solved by the EMS software are summarized in table VI:

	Eff.	μ_1	μ_2	μ_3	μ_4	v_1	v_2
Average	84%	.03	.01	.01	.02	.09	.18
Standard Dev.	19%	.03	.02	.02	.03	.08	.09
Coef. of Var.	.222	.99	1.53	1.98	1.31	.87	.51
Maximum	127%	.10	.10	.10	.10	.28	.32
Percentile 75%	97%	.06	.02	.01	.04	.14	.24
Percentile 50%	85%	.03	.00	.00	.01	.06	.20
Percentile 25%	70%	.00	.00	.00	.00	.03	.15
Percentile 4%	55%	.00	.00	.00	.00	.00	.00
Minimum	44%	.00	.00	.00	.00	.00	.00

Table VI- Statistical descriptors of the results for the EMS software (CCR model, minimization of inputs).

	Eff.	μ_1	μ_2	μ_3	μ_4	v_1	v_2
Average	74%	.035	.012	.007	.018	.074	.176
Standard Dev.	16%	.014	.005	.003	.011	.051	.064
Coef. of Var.	.221	.403	.421	.393	.633	.692	.367
Maximum	123%	.066	.024	.014	.042	.247	.276
Percentile 75%	90%	.045	.015	.008	.027	.098	.218
Percentile 50%	81%	.034	.011	.007	.017	.052	.192
Percentile 25%	65%	.023	.009	.005	.008	.042	.155
Percentile 4%	48%	.015	.003	.002	.000	.027	.035
Minimum	42%	.012	.002	.001	.000	.002	.000

Table VII- Statistical descriptors of the results for our new model (CCR model, minimization of inputs).

From the comparison between table VI and table VII we notice that:

The Statistical descriptors of the efficiency scores in our model are always lower. This was expected since penalties may occur and our global objective function never exceeds the traditional linear one.

The Average efficiency score in the classical CCR model is 84%, as was expected from the assumptions about the Half Normal distribution of the inefficiency.

The standard deviation of any multiplier is lower than the corresponding one in the non-adjusted case (this could originate from the decrease in the average values; that lead us also to compute the coefficient of variation, confirming that, even in relative terms, the weights are not as spread as in the original model.

The same conclusion also holds to any other measure (Maximum and percentiles).

Only when it comes to the minima of μ_4 and v_2 , we have a tie, but even in this case it is sufficient to take into account the values for the lower percentiles. In fact, μ_4 vanishes 35 times in the traditional linear model, in opposition to just 5 zeros in our model.

In the case of v_2 , the proportion is similar: there are only two null weights in the adjusted case against 13 on the classic one.

It should be noted that it would be easy to avoid the occurrence of null weights simply by increasing the steepness of the convex penalty function, although this was not our choice, since it could lead to a point that no DMU at all would reach the 100% efficiency score

In an effort to show our results in the 6 dimensional weights space, we illustrate in figures 11 through 13 its projections on the bidimensional spaces. In these figures, the circles represent the weights for the traditional model and the diamonds correspond to those obtained by our model.

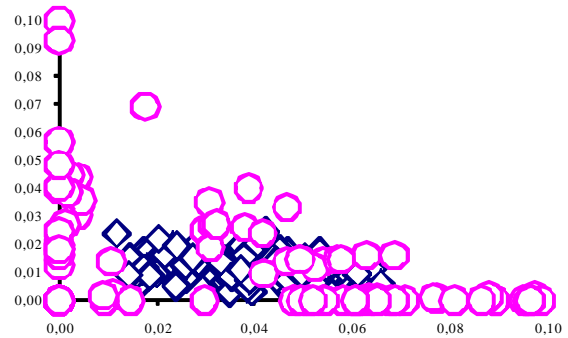


Figure 11- Distribution of μ_1 (in the vertical axis) and μ_2 (in the horizontal axis)

In figure 11, we can confirm that μ_1 and μ_2 cluster over the both axes, if we do not apply our adjustments on multipliers. In fact, μ_1 vanishes 31 times and about μ_2 this happens 43 times.

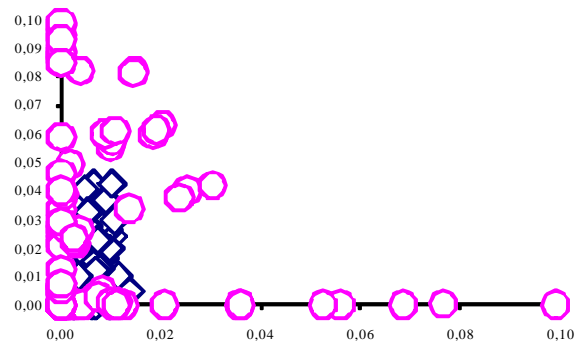


Figure 12- Distribution of μ_3 (in the vertical axis) and μ_4 (in the horizontal axis)

The considerations made about figure 11 also apply to figure 12 but here μ_3 has 38 zeros and μ_4 has 35. As a result of applying our technique only μ_4 kept some zeros, but nevertheless its number dropped to 5. It is worth mentioning that not only did we avoid the occurrence of zeros, but we also reduced its maximal values.

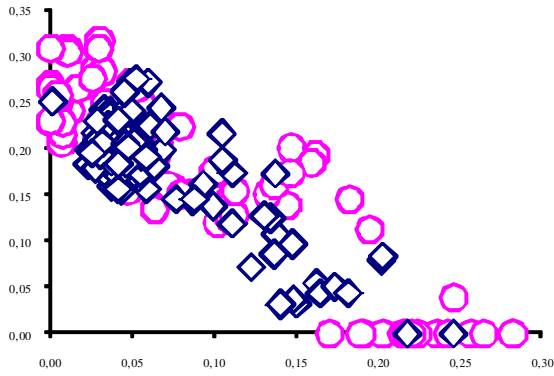


Figure 13- Distribution of v_1 (in the vertical axis) and v_2 (in the horizontal axis)

In figure 13, since we are dealing with only two outputs, the benefits of our method are not as evident as in the input case. Regardless of that, from the initial 13 DMUs that presented a null weight in output Y2, this number decreased to only 2 of the initial 13, namely DMU 41 and DMU 61.

If we had had more outputs, our technique would have been more useful, in the sense that the number of zeros to reduce would have been greater.

We tried to make this representation in several graphic ways, but this one seemed to us to be the best. We also investigated if the use of a data reduction technique like factor analysis, could be of some help, but the correlation matrix did not allow for an easy alternate representation of the data.

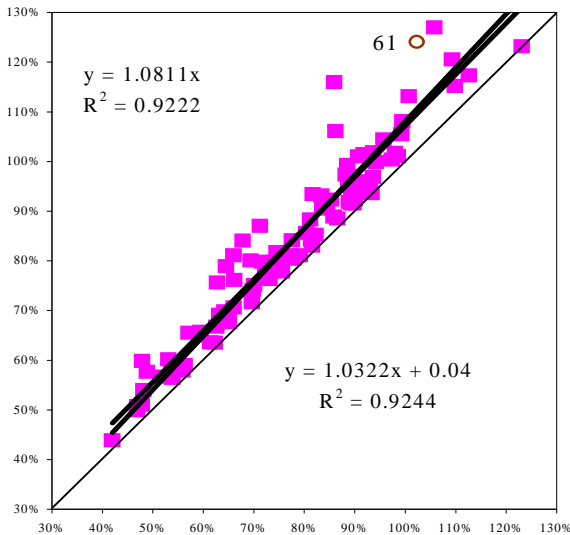


Figure 14- Representation of the Score obtained by the EMS software and our efficiency values

In figure 14, we plot the results for both the score obtained by the EMS software and our efficiency values. We notice that our values never exceed those from the CCR model and that in some cases there is a substantial reduction on the original value as is the case by instance

of DMU 61 whose efficiency value dropped from an original value of 124% to 102%.

Despite this fact, it is important to remark the existence of a strong correlation between the two variables, and so we made a linear regression on it.

Although the model with a constant leads to a higher determination coefficient of 0.9244, this constant has a p value of 0.11, and therefore, it is not significant. Thus we conclude that the EMS Scores exceed in 8% those from our model.

IX. CONCLUSIONS

This paper introduces a new way of adjusting weights, a matter that has already deserved many publications in the Data Envelopment Analysis field. This new method adds greater flexibility to the weight restrictions techniques. It is not our concern to present the potentialities or the details of weight restrictions. This matter is already extensively covered in the existing literature, namely on how to set up the specific values for the restrictions.

Since we are dealing with a non-linear concave objective function, we have the possibility to locate the optimal set of weights in a continuous way in contrast to the linear case where optimality always occurs at a vertex of the feasible solution set.

We used the simplest convex penalty function for the sake of clarity, but other convex functions like maximally flat polynomials or Tchebychev polynomials could also be used.

Even the convexity restriction can be dropped leading to more complicated programs, that can be used for instance in discriminating analysis.

REFERENCES

- [1] Milind Sathye (2003), "Efficiency of banks in a developing economy: The case of India". *European Journal of Operational Research*, 148, (3) 662-671
- [2] Thanassoulis, E. and P. Dunstan, (1994), "Guiding schools to improved performance using data envelopment analysis: An illustration with data from a local education authority", *Journal of the Operational Research Society*, 45, (11), 1247-1262.
- [3] Thanassoulis, E., (1993), "Comparison of regression analysis and data envelopment analysis as alternative methods for performance assessments", *Journal of the Operational Research Society*, 44, (11), 1129-44.
- [4] Boussofiane, A., R. G. Dyson, Thanassoulis, E., (1991), "Applied data envelopment analysis", *European Journal of Operational Research*, 52, (1), 1-15.
- [5] Charnes, A., Cooper, W. W., Rhodes, E., (1978), "Measuring the efficiency of decision making

- units”, *European Journal of Operations Research*, 2, (6), 429-44.
- [6] Farrell, M. J., (1957), “The measurement of productive efficiency”, *Journal of Royal Statistical Society A*, 120, 253-281.
- [7] Chang, K.-P. and Y.-Y. Guh, (1991), “Linear production functions and the data envelopment analysis”, *European Journal of Operational Research*, 52, (2), 215-23
- [8] Rhodes, E., (1978), “Data envelopment analysis and related approaches for measuring the efficiency of decision-making units with an application to program follow through in U.S. education”, unpublished doctoral thesis, School of Urban and Public Affairs, Carnegie - Mellon University.
- [9] Nunamaker, T. R. and A. Y. Lewin, (1983), “Measuring routine nursing service efficiency: A comparison of cost per patient day and data envelopment analysis models”, *Health Services Research*, 18, (2 (Part 1)), 183-208.
- [10] Thanassoulis, E., A. Boussofiene, R. G. Dyson, (1995), “Exploring output quality targets in the provision of prenatal care in England using data envelopment analysis”, *European Journal of Operational Research*, 80, (3), 588-607.
- [11] Ganley, J. A., Cubbin, J. S., (1992), *Public Sector Efficiency Measurement - Applications of Data Envelopment Analysis*, North - Holland, 57-80.
- [12] Lewin, A. Y. and R. C. Morey, Cook, T.J., (1982), “Evaluating the Administrative Efficiency of Courts”, *Omega*, 10, (4), 401-11.
- [13] Chu, X. and G. J. Fielding, (1992), “Measuring transit performance using data envelopment analysis”, *Transportation Research Part A (Policy and Practice)*, 26A, (3), 223-30.
- [14] Thompson, R. G., Dharmapala, P. S., Thrall, R., (1995), “Sensitivity analysis of efficiency measures with applications to Kansas farming and Illinois coal mining”, *Data Envelopment Analysis: Theory, Methodology and Applications.*, A. Charnes, W. W. Cooper, A. Lewin and L. M. Seiford. Boston, Kluwer Academic Publishers, 89-94.
- [15] Schefczyk, M., (1993), “Operational performance of airlines: an extension of traditional measurement paradigms”, *Strategic Management Journal*, 14, 301-317.
- [16] Vassiloglou, M. and D. Giokas, (1990), “A study of the relative efficiency of bank branches: an application of data envelopment analysis”, *Journal of the Operational Research Society*, 41, (7), 591-7.
- [17] Bessent, A., Bessent, W., Kennington, J., e Regan, B., (1982), “An application of Mathematical Programming to assess productivity in the Houston independent school district”, *Management Science*, Vol. 28, No. 12, 1355-67.
- [18] Charnes, A., Cooper, W.W., Rhodes, E., (1981), “Evaluating program and managerial efficiency: an application of Data Envelopment Analysis to Program Follow Through”, *Management Science*, 27, (6), 668-97.
- [19] Charnes, A., Cooper, W.W., e Rhodes, E., (1978), “Measuring the efficiency of decision making units”, *European Journal of Operational Research*, 2, (6), 429-44.
- [20] Castro, R., (1993), “O método DEA - Aplicação à avaliação da eficiência comparativa das escolas secundárias do distrito do Porto”, unpublished Msc. thesis, Faculdade de Economia, University of Oporto.
- [21] Santos, J., (1994), “Ordenação de unidades eficientes por técnicas de data envelopment analysis”, unpublished Msc. thesis, Instituto Superior Técnico, Technical University of Lisbon.
- [22] Beasley, J.E., (1990), “Comparing university departments”, *Omega*, 18, 171-83.
- [23] Charnes, A., Cooper, W.W., Rhodes, E., (1978), “Measuring the efficiency of decision making units”, *European Journal of Operational Research*, 2, (6), 429-44.
- [24] Charnes, A., Cooper, W. W., Rhodes, E., (1979), “Short communication: measuring the efficiency of decision making units”, *European Journal of Operational Research*, 3, (4), 339.
- [25] Charnes, A., Cooper, W.W., (1962), “Programming with linear fractional functionals”, *Naval Research Logistics Quarterly*, 9, 181-185.
- [26] Andersen, P., Petersen, N.C., (1993), “A procedure for ranking efficient units in data envelopment analysis”, *Management Science*, 39, (10), 1261-4.
- [27] Dyson, R. G. and E. Thanassoulis (1988). "Reducing weight flexibility in data envelopment analysis." *Journal of the Operational Research Society* **39**(6): 563-76
- [28] Roll, Y., W. Cook, et al. (1991). "Controlling factor weights in data envelopment analysis." *IIE Transactions* **23**(1): 2-9.
- [29] Wong, Y.-H. B. and J. E. Beasley (1990). "Restricting weight flexibility in data envelopment analysis." *Journal of the Operational Research Society* **41**(9): 829-35.
- [30] Thompson, R. G., F. D. Singleton, Jr., et al. (1986). "Comparative site evaluations for locating a high-energy physics lab in Texas." *Interfaces* **16**(6): 35-49.
- [31] Thompson, R. G., L. N. Langemeier, et al. (1990). "The role of multiplier bounds in efficiency analysis with application to Kansas farming." *Journal of Econometrics* **46**: 93-108.

- [32] Charnes, A., Cooper, W., Wei, Q., and Huang, Z. (1989), "Cone Ratio Data Envelopment Analysis and Multi-Objective Programming", *International Journal of Systems Science* **20**, 1099-1118.
- [33] Frederick S. Hillier, Gerald J. Lieberman, *Introduction to Operations Research*, Mc Graw Hill, 1990.
- [34] Durchholz, M. and R. Barr (1997). "Parallel and hierarchical decomposition approaches for solving large-scale data envelopment analysis models." *Annals of Operations Research*