Determining the Common Set of Weight in Data Envelopment Analysis with Linear Programs

Donna N. M. Sirait¹, Syahril Efendi², Sawaluddin³

¹Faculty of math and science, Universitas Sumatera Utara
²Faculty of computer science and information technology, Universitas Sumatera Utara
³Faculty of math and science, Universitas Sumatera Utara

* dehijo@gmail.com

Abstract. Data Envelopment Analysis (DEA) serves to assess efficiency in the use of resources (input) to achieve results (output) whose purpose is to maximize efficiency. DEA was first introduced by Charnes, Cooper dan Rhodes in 1978. DEA is a nonparametric approach which is basically the development of Linear Programming (LP). The characteristic of DEA CCR model is that it allows the DMU to measure the maximum efficiency value with the most favorable weight. The efficiency of the DMU obtained differs from the different sets of weight may not be comparable and rank on the same basis. In DEA model optional sets of input weight and output weight are usually assumed to represent the DMU that is considered the best to compare with all DMUs. This set of weights is usually different for each DMU. In this journal to determine the set of weights in the DEA is performed with a linear programs. After the linear program is determined, the value of the output weight and input weight is solved using the software POM-QM for Windows.

1. Introduction

Data Envelopment Analysis (DEA) was first introduced by Charnes, Cooper and Rhodes (CCR model) in 1978. DEA is a nonparametric approach which is basically the development of the Linear Program. DEA serves to assess efficiency in the use of resources (input) to achieve results (output) whose purpose is to maximize efficiency. DEA is a nonparametric method in operations and economic research to estimate production limits. This is used to empirically measure the Decision Making Unit (DMU) production efficiency.

The DEA has procedures that force individual weighting for each participating DMU, resulting in restrictions on DEA flexibility. To overcome the flexibility of DEA not to be restrained in determining the set of weights, it needs to be resolved through solving linear programming problems. The purpose of this study was to determine the set of weights in Data Envelopment Analysis with linear programs.

2. Research Method

The CCR model can be stated :

\[
\text{Max } \sum_{i=1}^{t} u_i y_{ip}
\]
Refers to:
\[ \sum_{i=1}^{m} v_i x_{ip} = 1 \]  
\[ \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \quad \forall j, \quad u_r \geq \varepsilon, v_i \geq \varepsilon, \forall i, r, \]

Where \( u_r, (r = 1, \ldots, s) \) dan \( v_i, (i = 1, \ldots, m) \) which is the weights that associated with each of the \( r \) output and input \( i \) and \( \varepsilon \) is the very small epsilon.

The CCR model with the assumption of limits on factor weight is:
\[
\text{Max} \sum_{r=1}^{s} u_r y_{rp} \]

Refers to:
\[ \sum_{i=1}^{m} v_i x_{ip} = 1 \]  
\[ \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \quad \forall j, \]
\[ U_r^1 \leq U_r \leq U_r^u \quad \forall r, \]
\[ V_i^1 \leq V_i \leq V_i^u \quad \forall i, \]

Where \( U_r^1, U_r^u, V_i^1 \) dan \( V_i^u \) is the upper and lower limits of each input and output weights.

To determine the upper limit of the output weight, the following problems are considered:
\[
\text{Max} u_r \]

Refers to:
\[ \sum_{i=1}^{m} v_i x_{ip} \leq 1 \]  
\[ \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \quad \forall j, u_r \geq 0 \forall i, r. \]

A similar problem is solved to determine the upper limit of input weight. In equation (3), the maximum value of each factor weight is determined in such a way that the efficiency of each DMU does not exceed 1. \( \sum_{i=1}^{m} v_i x_{ip} \leq 1 \) is normalizing constraints and normalizing factors. The upper limit of the output and input weights is determined by solving the \( s + m \) linear programming problem. It is important to prove the feasibility of the problem (3) and, the limitations and positivity of the optimal value, because in the absence of each of these assumptions; the proposed procedure will be split.

**Theorem 1.** Problems (3) are feasible and their optimal values are limited and positive.

**Proof.** Obviously \((U, V) = (0, 0)\) is a viable solution of (3). To prove the limits of (3), constraints can be considered \( \sum_{i=1}^{m} v_i x_{ij} \leq 1 \) and \( \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \). This concludes that \( \sum_{r=1}^{s} u_r y_{rj} \leq 1 \).

For every \( p (1 \leq p \leq s) \) there is at least one \( y_{pq} (1 \leq q \leq s) \) such that \( y_{pq} \neq 0 \). Therefore:

\[
u_p \leq (1 - \sum_{r=1}^{s} u_r y_{rq})/y_{pq} \]
It can be concluded that the problem of determining the upper limit of the output weight is limited. In the same way, it can be proved that the upper limit for input weights is also limited. Consider the multiplicity of the upper limit for the problem of output weight:

$$\min \sum_{j=1}^{n} \alpha_j$$

Refers to:

$$\sum_{j=1}^{n} \alpha_j x_{ij} - \sum_{j=1}^{n} \beta_j x_{ij} \geq 0 \forall i,$$

$$\sum_{j=1}^{n} \beta_j y_{rij} \geq 0 \forall r \neq p,$$

$$\sum_{j=1}^{n} \beta_j y_{pi} \geq 1$$

$$\alpha_j \beta_j \geq 0 \quad \forall j$$

Suppose the optimal value of the upper limit for the problem of output weight is zero. So that with a strong property duality, \( \sum_{j=1}^{n} \alpha_j^* = 0 \) or \( \alpha_j^* = 0 (j = 1, ..., n) \). Consider obstacles \( \sum_{j=1}^{n} \alpha_j x_{ij} - \sum_{j=1}^{n} \beta_j x_{ij} \geq 0 (i = 1, ..., n) \), yields that \( \sum_{j=1}^{n} \beta_j y_{rij} \geq 0 (i = 1, ..., n) \) and this is a contradiction. Thus, the optimal value of (3) is positive.

Because (3) has a special structure, so that the optimal value can be obtained by comparing the constraints. In (3) it is clear that \( \sum_{r=1}^{s} u \cdot y_{rij} \leq 1 (j = 1, ..., n) \). Therefore the upper limit of the input and output weights can be calculated as follows:

$$u_r^* = 1/ \max_{1 \leq j \leq s} y_{rij} \text{ for } (r = 1, ..., s)$$

$$v_i^* = 1/ \max_{1 \leq j \leq n} x_{ij} \text{ for } (i = 1, ..., n)$$

**Step 2.** Determine a set of weights

Starting from the restricted model (2), the set of weights can be achieved by expressing deviations from the partial limits of the range between the upper and lower boundary weights. Assuming the same deviation from the limits in all DMUs, we get:

Max \( \emptyset \)

Mengacu pada:

$$\sum_{r=1}^{s} u \cdot y_{rij} \cdot \sum_{i=1}^{n} v_i x_{ij} \leq 0 \forall j,$$

$$U_r^i + \emptyset (U_r^u - U_r^l) \leq u_r \leq U_r^u - \emptyset (U_r^u - U_r^l) \forall r,$$

$$V_i^l + \emptyset (V_i^u - V_i^l) \leq v_i \leq V_i^u - \emptyset (V_i^u - V_i^l) \forall i,$$

Compaction of the weight of the intervals in (6) is carried out with the respective proportions of the interval length, because the upper limit of the factor weight is not the same. Applying (5) and setting the lower limit of the factor weight equals zero the results in the following model:

Max \( \emptyset \)

Refers to:
\[
\sum_{r=1}^{s} u_{r} y_{rj} = \sum_{i=1}^{m} v_{i} x_{ij} \leq 0 \forall j, \tag{7}
\]

Where \( U_{r}(r = 1, ..., s) \) and \( V_{i}(i = 1, ..., m) \) calculated by (5).

**Theorem 2.** Problem (7) is feasible and its optimal value is limited and positive.

**Proof.** The proof is similar to the theorem 1.

A set of weights is obtained by solving (7) and the efficiency of each DMU can be evaluated as follows:

\[
e_{j} = \frac{\sum_{r=1}^{s} u_{r} y_{rj}}{\sum_{i=1}^{m} v_{i} x_{ij}} \forall j, \tag{8}
\]

Where \( u_{r}^{*}(r = 1, ..., s) \) and \( v_{i}^{*}(i = 1, ..., m) \) is the optimal values of (7).

In cases where none of the DMUs are efficient, all output weights can be increased (and / or the input weight decreases) by minimal proportions until an efficient DMU is achieved. One way to do a task is to follow the following substitutions:

\[ M_{r} = \frac{u_{r}^{*}}{e}, N_{i} = v_{i}^{*}, \quad \forall r, i, \]

Where \( e = \max_{1 \leq j \leq m} e_{j} \). The resulted weights \( M_{r}(r = 1, ..., s) \) and \( N_{i}(i = 1, ..., m) \) are the proposed of the common set weight.

After raising the weight set, the efficiency of the DMU is determined by:

\[
e_{j} = \frac{\sum_{r=1}^{s} M_{r} y_{rj}}{\sum_{i=1}^{m} N_{i} x_{ij}} \forall j, \tag{9}
\]

3. Numerical example

In this opportunity, the author will use data from 15 laptop brands with different specifications using 2 input variables and 3 output variables without considering the efficiency of the processing results.

The output data used in this research is the battery life as long as the laptop is turned on, RAM and the capacity of the hard disk drive (HDD) on the laptop.

**Table 1.** Data from 15 types of laptops

| Laptop brand (model) | Production Cost ($)| Weight (Kg) | Battery Life (Hours) | RAM (Gb) | HDD Capacity (GB) |
|----------------------|-------------------|------------|---------------------|----------|------------------|
| HP (11-E010nr)       | 414.99            | 1.54       | 5                   | 4        | 500              |
| Acer (V5-131)        | 339.98            | 1.50       | 6.5                 | 4        | 500              |
| Acer (AS1410)        | 449.99            | 1.41       | 6                   | 2        | 250              |
| Lenovo (X100e)       | 498.88            | 1.50       | 7.5                 | 1        | 250              |
| Acer (V5-171)        | 683.67            | 1.36       | 5                   | 6        | 500              |
| ASUS (X200CA)        | 299.99            | 1.18       | 5                   | 2        | 500              |
| ASUS (Q200E)         | 399.99            | 1.41       | 11                  | 4        | 500              |
| Lenovo (IdeaPad S210)| 279.99            | 1.41       | 4                   | 4        | 500              |
| Dell (Inspiron 11)   | 299.99            | 1.41       | 8                   | 2        | 500              |
| Acer (S3-391-6046)   | 405.00            | 1.36       | 3                   | 4        | 320              |
| Gateway (LT410P04u)  | 338.45            | 1.09       | 5                   | 2        | 320              |
| ASUS (1015E)         | 259.00            | 1.27       | 7.5                 | 2        | 320              |
Based on the data that will be processed, if it is exemplified in the linear program formulation where we take Lenovo (IdeaPad S210) as a primal model, it can be written:

DMU 8 Lenovo (IdeaPad S210) as Primal Model:

Max Z = 4 X₁ + 4 X₂ + 500 X₃

Constraints:

279.99 Y₁ + 1.41 Y₂ = 1
5 X₁ + 4 X₂ + 500 X₃ − 414.99 Y₁ − 1.54 Y₂ ≤ 0
6,5 X₁ + 4 X₂ + 500 X₃ − 339.98 Y₁ − 1.50 Y₂ ≤ 0
6 X₁ + 2 X₂ + 250 X₃ − 449.88 Y₁ − 1.41 Y₂ ≤ 0
7,5 X₁ + X₂ + 250 X₃ − 498.88 Y₁ − 1.50 Y₂ ≤ 0
5 X₁ + 6 X₂ + 500 X₃ − 683.67 Y₁ − 1.36 Y₂ ≤ 0
5 X₁ + 2 X₂ + 500 X₃ − 299.99 Y₁ − 1.18 Y₂ ≤ 0
11 X₁ + 4 X₂ + 500 X₃ − 399.99 Y₁ − 1.41 Y₂ ≤ 0
8 X₁ + 2 X₂ + 500 X₃ − 299.99 Y₁ − 1.41 Y₂ ≤ 0
3 X₁ + 4 X₂ + 320 X₃ − 405 Y₁ − 1.36 Y₂ ≤ 0
5 X₁ + 2 X₂ + 320 X₃ − 338.45 Y₁ − 1.09 Y₂ ≤ 0
7,5 X₁ + 2 X₂ + 320 X₃ − 259 Y₁ − 1.27 Y₂ ≤ 0
4,5 X₁ + 4 X₂ + 500 X₃ − 669.58 Y₁ − 1.36 Y₂ ≤ 0
9 X₁ + 2 X₂ + 250 X₃ − 300 Y₁ − 1.59 Y₂ ≤ 0
5,45 X₁ + 2 X₂ + 320 X₃ − 340 Y₁ − 1.45 Y₂ ≤ 0
Y₁, Y₂, X₁, X₂, X₃ ≥ 0

Y₁ = CSW for input production cost
Y₂ = CSW for input weight
X₁ = CSW for output battery life
X₂ = CSW for output RAM
X₃ = CSW for output HDD capacity

Completion of the linear program formulation above is done using the help of POM-QM software for Windows and the results can be seen in the table below.

### Table 2. Input and output weights with DMU 8 as primal models

| DMU | INPUT | OUTPUT |
|-----|-------|--------|
|     | Y₁ = 0.004 | X₁ = 0 |
| DMU 8 | Y₂ = 0 | X₂ = 0.071 |
|     |         | X₃ = 0.002 |

In the same way the calculation is done by taking each DMU as a primal model. So that the results are obtained as shown in the table below.

### Table 3. The set of weight for each DMU

| DMU | CCR | Input Weight | Output eight |
|-----|-----|--------------|--------------|
|     |     | Y₁ | Y₂ | X₁ | X₂ | X₃ |
| D01 | 0.883 | 0.000 | 0.544 | 0.007 | 0.062 | 0.001 |
| D02 | 0.950 | 0.003 | 0.000 | 0.050 | 0.071 | 0.001 |
| D03 | 0.545 | 0.000 | 0.709 | 0.091 | 0.000 | 0.000 |
| D04 | 0.641 | 0.000 | 0.667 | 0.085 | 0.000 | 0.000 |
| D05 | 1.000 | 0.000 | 0.735 | 0.000 | 0.250 | 0.000 |
4. Conclusion

In this journal, the author presents the DEA model with the set of weights being the Linear Programming model. Using this method, linear programming problems are solved by software POM-QM for Windows. After solving the model, it can be determined the efficiency of each DMU and the set of weights in the DEA obtained. So that the solution of this journal can be useful and used for subsequent studies.

References

[1] Daavoodi, A dan H. Zhiani Rezai. 2011. Common set of weights in data envelopment analysis: a linear programming problem.
[2] A. Wanto, M. Zarlis, Sawaluddin, and D. Hartama, “Analysis of Artificial Neural Network Backpropagation Using Conjugate Gradient Fletcher Reeves in the Predicting Process,” *Journal of Physics: Conference Series*, vol. 930, no. 1, pp. 1–7, 2017.
[3] Kao, C dan H.T. Hung. 2005. Data envelopment analysis with common weights: the compromise solution approach. Journal of the Operational Research Society. Taiwan.
[4] B. Febriadi, Z. Zamzami, Y. Yunefri, and A. Wanto, “Bipolar function in backpropagation algorithm in predicting Indonesia’s coal exports by major destination countries,” *IOP Conference Series: Materials Science and Engineering*, vol. 420, no. 12089, pp. 1–9, 2018.
[5] Pin Bao, Chiao. Nai-Chien Wei dan Meei-Ing Tsai. 2016. Constructing Reasonable Common Weights In Data Envelopment Analysis. International Journal of Organizational Innovation Vol 8 Num 3.
[6] N. Nasution, A. Zamsuri, L. Lisnawita, and A. Wanto, “Polak-Ribiere updates analysis with binary and linear function in determining coffee exports in Indonesia,” *IOP Conference Series: Materials Science and Engineering*, vol. 420, no. 12089, pp. 1–9, 2018.
[7] Sun, Jiasen. Jie Wu dan Dong Guo. 2013. Performance ranking of unit considering ideal and anti ideal DMU with common weight. Applied Mathematical Modelling