Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
April 2007, Vol. 3, No. 4, 67-70

A new approach to determine efficient DMUs in DEA models using inverse optimization

Gholam R. Amin*

Assistant Professor of OR, Dep. of Computer Science, Postgraduate Engineering Centre, Islamic Azad University, Tehran South Branch, Tehran, Iran

Abstract

This paper proposes a new approach for determining efficient DMUs in DEA models using inverse optimization and without solving any LPs. It is shown that how a two-phase algorithm can be applied to detect efficient DMUs. It is important to compare computational performance of solving the simultaneous linear equations with that of the LP, when computational issues and complexity analysis are at focus.

Keywords: Data Envelopment Analysis (DEA); Decision Making Units (DMUs); Inverse optimization; Ellipsoid algorithm

1. Introduction

The DEA model is a programming technique for the construction of a non-parametric, piecewise linear convex hull to the observed set of input and output data for discussions of methodology [5,6]. DEA defines a linear segmentation to envelop the whole sample data, and uses radial expansion or concentration to measure the efficiency [2]. This methodology, proposed initially by Charnes, Cooper and Rhodes and known as CCR model. An inverse optimization problem consists of inferring the values of the model parameters such as cost coefficient, right hand side vector, and the constraint matrix given the values of observable parameters (optimal decision variables) [1]. Geophysical scientists were the first ones in studying inverse optimization problems. In the early few years, inverse optimization problems attracted many operation research specialists and different kinds of inverse optimization problems have been studied by researchers. In this paper, a new approach to determine efficient DMUs in DEA models based on inverse optimization under L_1 norm, without solving any LPs is delivered. For this end, a two-phase polynomial time algorithm is proposed. Phase I solves a system of linear equations. In order to show that the corresponding DMU is efficient or not, Phase II applied on the outcomes of the Phase I by using an ellipsoid algorithm.

Amin and Toloo [3] proposed a polynomial-time algorithm for computing the non-Archimedean ε in DEA models, which there is no need to identify the specific value of ε in this paper.

2. A necessary and sufficient condition for efficient DMUs

Let **S** denote the set of feasible solutions for an optimization problem called as *P*, the relevant specified cost vector is **c**, and x^0 be a given feasible solution. The inverse optimization problem is to perturb the cost vector **c** to **d**, so that x^0 is an optimal solution of *P* with respect to **d** and $\|d-c\|_p$ is minimum, where $\|d-c\|_p$ is some selected L_P norm. Consider the following linear programming:

$$\text{Min } \sum_{j=1}^{n} c_j x_j$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad i=1,2.....m, \tag{1}$$

$$x_j \geq 0 \qquad j=1,2.....n.$$

* *Corresponding author. E-mail: G_Amin@azad.ac.ir*

Suppose that x^0 be a feasible solution. The corresponding inverse problem under L_1 norm is as follows [1]:

$$\text{Min} \sum_{j=1}^{n}(\alpha_j+\beta_j)$$

Subject to:

$$\sum_{i=1}^{m} a_{ij}\pi_i-\alpha_j+\beta_j+\gamma_j=c_j \qquad \forall j\in L,$$

$$\sum_{i=1}^{m} a_{ij}\pi_i-\alpha_j+\beta_j=c_j \qquad \forall j\in F, \qquad (2)$$

$$\alpha_j\ge 0,\ \beta_j\ge 0 \qquad j=1,\dots,n,$$

$$\gamma_j\ge 0 \qquad \forall j\in L,\ \pi_i free,\ i=1,\dots,m,$$

where $L=\{j: x_j^0=0\}$, $F=\{j:0<x_j^0\}$. In the basic DEA Models, the k^{th} DMU is obviously efficient if and only if $\theta^*=1$ and all the slack variables are equal to zero. Notice that for the k^{th} DMU the objective function in CCR and BCC model is $z_k=\theta-\varepsilon(\mathbf{1s}^i+\mathbf{1s}^o)$. Now consider the feasible solution $\mathbf{x}^0=(\theta,\boldsymbol{\lambda},\mathbf{s}^i,\mathbf{s}^o)$ with $\theta=1,\lambda_k=1$ and $\lambda_j=0$ for all $j=1,\dots.,n,\ j\ne k$, $\mathbf{s}^i=\mathbf{0}$ and $\mathbf{s}^o=\mathbf{0}$ in the CCR model. The corresponding inverse linear program for the k^{th} DMU is as follows:

$$\text{Min} \sum_{j=1}^{n+m+s+1}(\alpha_j+\beta_j)$$

Subject to:

$$\sum_{i=1}^{m+s} a_{ij}\pi_i-\alpha_j+\beta_j+\gamma_j=c_j \qquad \forall j\in L,$$

$$\sum_{i=1}^{m+s} a_{ij}\pi_i-\alpha_j+\beta_j=c_j \qquad \forall j\in F, \qquad (3)$$

$$\alpha_j\ge 0,\ \beta_j\ge 0 \qquad j=1,\dots,n+m+s+1,$$

$$\gamma_j\ge 0 \qquad \forall j\in L,\ \pi_i free,\ i=1,\dots,m+s,$$

where n is the number of DMUs, m and s are the number of inputs and outputs respectively, and

$$L=\{2,\dots,k,k+2,\dots,m+n+s+1\},\ F=\{1,k+1\}.$$

Notice that for each $j\in L,\ c_j\in\{0,-\varepsilon\}$ and for each $j\in F,\ c_j\in\{0,1\}$. It is easy to see that if the optimal value of the inverse problem is equal to zero then x^0 also is an optimal solution of the CCR model. Now consider the essential theorem given in Section 3.

3. The essential theorem

Theorem 1. The k^{th} DMU is efficient if and only if the following simultaneous linear equations have a solution:

$$\sum_{i=1}^{m+s} a_{ij}\pi_i+\gamma_j=c_j \qquad \forall j\in L,$$

$$\sum_{i=1}^{m+s} a_{ij}\pi_i=c_j \qquad \forall j\in F, \qquad (4)$$

$$\gamma_j\ge 0 \qquad \forall j\in L,\ \pi_i free,\ i=1,\dots,m+s.$$

Proof.

Sufficient Condition. Suppose that the above system has a solution, Say, (π^0,γ^0), then by taking $\alpha_j^0=\beta_j^0=0$ for each $j=1,\dots.,m+n+s+1$, $(\pi^0,\gamma^0,\alpha^0,\beta^0)$, is an optimal solution of the inverse problem. Therefore x^0 is an optimal solution of CCR model, so the k^{th} DMU is efficient.

Necessary Condition. Conversely, suppose that the k^{th} DMU is efficient then x^0 is an optimal solution and the corresponding inverse LP has the zero optimal solution value, that is, $\alpha_j^*=\beta_j^*=0$ for each $j=1,\dots.,m+n+s+1$. So the constraint of the inverse LP must has a solution with $\alpha_j=\beta_j=0$ (for each j). The mentioned proof clarifies the necessary condition.

Notice that the only difference is $\sum_{i=1}^{m+s+1} a_{ij}\pi_i$ that appears instead of $\sum_{i=1}^{m+s} a_{ij}\pi_i$ in the equations and all other details are the same, if the above Theorem is applied for the BCC model.

4. The algorithm

For each $j = n+2, \ldots, n+m+s+1$, the corresponding equations stated in (4) can be reduced as:

$$\sum_{i=1}^{m+s} a_{ij}\pi_i + \gamma_j = -\pi_{j-(n+1)} + \gamma_j = -\varepsilon. \qquad (5)$$

Because the Non-Archimedean $\varepsilon > 0$, hence in every feasible solution of (4), $\pi_i > 0,\ i = 1, \ldots, m+s$. In the following, a two-phase algorithm is proposed to detect (4) has a solution or not. Phase I solves the following linear equations:

$$\begin{aligned} &\sum_{i=1}^{m} x_{ik}\pi_i = 1, \\ &-\sum_{i=1}^{m} x_{ik}\pi_i + \sum_{r=1}^{s} y_{rk}\pi_{m+r} = 0, \\ &-\sum_{i=1}^{m} x_{ij}\pi_i + \sum_{r=1}^{s} y_{rj}\pi_{m+r} + \gamma_j = 0, \\ &j = 2, \ldots, n+1, j \neq k+1. \end{aligned} \qquad (6)$$

One can see easily that, the system (4) has a solution if and only if there is a solution for (6) with the all of positive π_i. Note that there are $(n+1)$ equations and $(n+m+s-1)$ free variables in (6). So, Phase I concludes at most $(m+s-2)$ independent variables. The output of the Phase I may be denoted as $y = b - Ax$, where x, y present the independent and dependent variables respectively, without loss of generality. Phase II takes the general solution of (6) which is obtained from Phase I to produce a positive solution (if there is any). For this aim an ellipsoid Algorithm is used to detect that $S = \{x : g - Gx > 0\}$ has a solution or it is empty, where $\mathrm{G} = \begin{pmatrix} \mathrm{A} \\ \text{-}\,\mathrm{I} \end{pmatrix}$ is $r \times q$, $g = \begin{pmatrix} b \\ 0 \end{pmatrix}$ is $r \times 1$, $r \geq 2$ and $q \geq 2$ [4]. Notice $g - Gx > 0$ implies that $b > Ax,\ x > 0$. Let L be the size of $Gx < g$ that is, the number of bits are needed to present it. The two-phase algorithm is:

Phase I ;

begin

Use an elimination method to derive $y = b - Ax$ from (6);

end

Phase II ;

begin

If there is a row $y_i = b_i - a^i x$ that has the all non positive coefficients, then return $S = \phi$ and stop;

If $b > 0$ or there is a column in $-A$ that has the all positive coefficients, then return $S \neq \phi$ and stop;

Step 1. set $k = 0,\ x_0 = 0$ and $\mathrm{P}_0 = q^2 2^{2L} \mathrm{I}_{q \times q}$.

Step 2. If $x_k \in S$, then return k and stop. If $k > 16q(q+1)L$, then return $S = \phi$ and stop.

Step 3. Choose an inequality $G_v x_k \geq g_v$ violated by x_k and set:

$$x_{k+1} = x_k - \frac{1}{1+q} \frac{\mathrm{P}_k G_v^t}{\sqrt{G_v \mathrm{P}_k G_v^t}},$$

$$Q_{k+1} = \frac{q^2}{q^2 - 1}\left[Q_k - \frac{2}{(1+q)} \frac{(\mathrm{P}_k G_v^t)(\mathrm{P}_k G_v^t)^t}{G_v \mathrm{P}_k G_v^t}\right],$$

and go to step 2.

end

Theorem 2. The two-phase algorithm runs in $O(\max\{q^4 L, (n+1)^3\})$.

Proof. It is obviously clear that Phase I runs in $O((n+1)^3)$. Because there are $O((n+1)^2)$ iterations and $O(n+1)$ operations in each iteration. On the other hand, Phase II runs in $O(q^4 L)$ [4]. This completes the proof.

5. Illustrated example

Suppose on a given system there are two DMUs, three inputs and one output such as table 1.

Table 1. Example data.

DMU No.	I_1	I_2	I_3	O
1.	2	3	4	1
2.	2	3	5	1

Obviously with respect to the first DMU, the second one is inefficient. Consider the second DMU. Phase I solves the following linear system.

$$\begin{aligned} &2\pi_1 + 3\pi_2 + 5\pi_3 = 1, \\ &-2\pi_1 - 3\pi_2 - 5\pi_3 + \pi_4 = 0, \\ &-2\pi_1 - 3\pi_2 - 4\pi_3 + \pi_4 + \gamma_3 = 0. \end{aligned} \tag{7}$$

The general solution is:

$$\pi_1 = \frac{1}{2} - \frac{3}{2}\pi_2 + \frac{5}{2}\gamma_3,$$

$$\pi_3 = -\gamma_3,$$

$$\pi_4 = 1.$$

Note that there is a row $(0,0,\text{-}1)$ corresponding to the second equation which is not positive. According to the algorithm the second DMU is not efficient. Now consider the first DMU, according to the algorithm this one is efficient if and only if the following system has a solution.

$$\begin{pmatrix} \frac{3}{2} & 2 \\ 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \pi_2 \\ \gamma_2 \end{pmatrix} < \begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \end{pmatrix}$$

Let $x_0 = \left(\frac{1}{4}, \frac{2}{5}\right)$ and $Q_0 = \mathrm{I}$ (for convenience) Phase II gives the feasible solution $x_0 = \left(\frac{1}{20}, \frac{11}{15}\right)$ after the first iteration. Therefore DMU1 is efficient.

6. Conclusion

Determining the most efficient DMUs in data envelopment analysis models requires solving the relevant linear programs. In this paper it is shown that by using the inverse optimization technique there is no need to solve any linear programs. A two-phase polynomial time algorithm is proposed. Phase I solves a system of linear equations and then an ellipsoid algorithm is applied in Phase II in order to the corresponding DMU is an efficient or not. A necessary and sufficient condition proved this hypothesis that the k^{th} DMU is efficient if and only if the relevant mentioned linear equations set has a solution. Our proposed approach is important to compare computational performance, when computational issues and complexity analysis are at focus.

References

[1] Ahuja, R. K. and Orlin, J. B., 2001, Inverse optimization. *Operations Research*, 49, 771-783.

[2] Ali, A. I., Lerme, C. S. and Seiford, L. M., 1995, Components of efficiency evaluation in Data Envelopment Analysis. *European Journal of Operational Research*, 80, 462-473.

[3] Amin, Gholam R. and Toloo, M., 2004, A polynomial-time algorithm for finding ε in DEA models. *Computers and Operations Research*, 31(5), 803-805.

[4] Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D., 2005, *Linear Programming and Network Flows*. Third Edition, John Wiley & Sons, New York.

[5] Charnes, A., Cooper, W. W., Lewin, A. Y. and Seiford, L. M., 1994, *Data Envelopment Analysis: Theory, Methodology and Application.* Boston, Kluwer Academic.

[6] Cooper, W. W., Seiford, L. M. and Tone, K., 2006, *Introduction to Data Envelopment Analysis and Its Uses with DEA-Solver Software and References.* Springer.