REMOVABLE OF LIMITATION DATA

Bikram kumar
P G Deptt of Mathematics,
TMBU, Bhagalpur (Bihar)

Abstract

In this paper removes a limitation of Imprecise Data Envelopment Analysis (IDEA) and Assurance Region – Data Envelopment Analysis (AR- IDEA) which requires access to actually attained maximum values in the data**.** This is accomplished by introducing a dummy variable that supplies needed normalizations on maximal values and this is done in a way that continues to provide linear programming equivalents to the original problems. This dummy variable can be regarded as a new Decision Making Unit (DMU), referred to as a Column Maximum DMU (CMD). The models and methods used are directed to deterministic uses of DEA. They are not intended to cover stochastic approaches to DEA as in the chance constrained programming formulations of Olesen and Petersen or the statistical characterizations provided by Banker . They are also not directed to potential data imprecisions such as are associated with the sensitivity of DEA results to data variations as in Seiford and Zhu . It is important to note that the nonlinear representations according to DEA in IDEA and AR-IDEA models are transformed into linear programming equivalents. In this paper, we have reviewed in brief the plan of IDEA and AR-IDEA given by Cooper, Park and Yu. A general and rigorously established form has been given to compute this paper.

Keywords: - Imprecise Data Envelopment Analysis (IDEA) , Assurance Region – Data Envelopment Analysis (AR- IDEA) , linear programming.

1.0 Introduction

Imprecise Data Envelopment Analysis (IDEA) extends DEA so it can simultaneously treat exact and imprecise data where the latter are known only to obey ordinal relations or to lie within prescribed bounds. AR-IDEA extends this further to include Assurance Region (AR) and the like approaches to constraints on the variables. In order to provide one unified approach, a further extension also includes cone-ratio envelopment approaches to simultaneous transformations of the data and constraints on the variables. The present paper removes a limitation of IDEA and AR-IDEA which requires access to actually attained maximum values in the data. This is accomplished by introducing a dummy variable that supplies needed normalizations on maximal values and this is done in a way that continues to provide linear programming equivalents to the original problems. This dummy variable can be regarded as a new Decision Making Unit (DMU), referred to as a Column Maximum DMU (CMD).

Cooper, Park and Yu [7] showed how DEA could be extended to treat not only exact data but also data that are known only ordinally or within prescribed bounds. The transformationsused make it possible to apply DEA to any combination of exact, ordinal or bounded data. It was also shown how conditions on the variables as well as the data could be treated in this same manner. This included (i) Assurance Region (AR) conditions on the variables, as in Thompson et.al. [12, 13] and (ii) the combined variable-data transformations employed in the cone-ratio envelopment of Charnes et al. [4] See also Brockett et.al. [2]. The resulting approaches are referred to as IDEA and AR-IDEA. A further extension was effected to treat strict, as well as weakly ordered, data in order to deal with applications of these ideas to the Korean mobile telecommunication company that is reported in Cooper, Park and Yu [8].

The models and methods used are directed to deterministic uses of DEA. They are not intended to cover stochastic approaches to DEA as in the chance constrained programming formulations of Olesen and Petersen [9, 10] or the statistical characterizations provided by Banker [1]. They are also not directed to potential data imprecisions such as are associated with the sensitivity of DEA results to data variations as in Seiford and Zhu [11].

It is important to note that the nonlinear representations accorded to DEA in IDEA and AR-IDEA models are transformed into linear programming equivalents. This is accomplished by rescaling the data and employing a variable alteration technique. In these variable alterations and rescalings, it is assumed that there exists at least one DMU which has a value that is maximal in the corresponding input or output data column.

Thrall, who served as a referee of Cooper, Park and Yu [7], called attention to the need for making this assumption explicit in order to display it as a possible limitation to potential uses of the approaches described in Cooper, Park and Yu [7]. It was felt (at least by us) that resolution of this problem might lead to nonlinear formulations analogous to those encountered in the attempt to extend AR approaches for treating linked-cone DEA assurance region approaches to the treatment of profit ratios like those described in Thompson, Dharmapala and Thrall [12]. However, as we shall show in this paper, it is possible to introduce a dummy variable in a way that makes it possible to obtain a linear programming equivalent in the manner described for IDEA and AR-IDEA in the original Cooper, Park and Yu formulations.

The plan of development is as follows. First, IDEA and AR-IDEA as given in Cooper, Park and Yu [7] are reviewed in brief. We then present a numerical example to show what is needed in concrete detail. We use this same example to show how the needed development can be accomplished by the simple expedient of introducing a dummy variable which we associate with a new DMU. The final section then puts this in a general and rigorously established form to complete this paper.

1.1 AR-IDEA model

Our present focus is on the AR-IDEA model which we write in the following form:

$$\left.\begin{aligned} &\max \pi_0 = \sum_{r=1}^{s} \pi_r y_{r0} \\ &\text{s.t.} \qquad \sum_{r=1}^{s} \pi_r y_{rj} - \sum_{i=1}^{m} \omega_r x_{ij} \le 0,\ j = 1,\dots,n \\ &\qquad\qquad \sum_{i=1}^{m} \omega_i x_{i0} = 1 \end{aligned}\right\} \quad (1.1a)$$

$$\left.\begin{aligned} &y_r = \left(y_{rj}\right) \in D_r^{+} r = 1,\dots,s \\ &x_i = \left(x_{ij}\right) \in D_r^{-} i = 1,\dots,m \end{aligned}\right\} \quad (1.1b)$$

$$\left.\begin{aligned} &\mu = \left(\mu_r\right) \in A^{+} \\ &\omega = \left(\omega_i\right) \in A^{-} \end{aligned}\right\} \quad (1.1c)$$

Here, y_{rj}, x_{ij}, respectively, represent the observed or recorded amounts of the rth output (r = 1, …, s) and the ith input (i = 1, …, m) for each DMU$_j$ (j = 1, …, n). The y_{r0}, x_{i0} data represent the outputs and inputs for DMU$_0$, the DMU$_j$ to be evaluated.

Now suppose that the data are not known exactly, so the sets D_r^+, D_r^- in (1.1b) represent imprecise data for the vector of output variables $y_r = (y_{rj}) = (y_{r1}, \ldots, y_{rn})$ and input variables $x_i = (x_{ij}) = (x_{i1}, \ldots, x_{in})$. As examples, for D_r^+ we could have

bounded data: $$D_r^+ = \left\{(y_{rj}) := (y_{rj}^-) \le y_{rj} \le y_{rj}^+,\ j = 1, \ldots, n\right\} \quad (1.2a)$$

ordinal data: $$D_r^+ = \left\{(y_{rj}) : y_{rj} \le y_{rj+1},\ j = 1, \ldots, n-1\right\} \quad (1.2b)$$

where y_{rj}^-, y_{rj}^+ are positive constants. D_i^- are defined in an analogous manner for inputs.

Turning to variables, the sets A^+, A^- in (1c) represent AR bounds on the multipliers $\mu = (\mu_r)$, $\omega = (\omega_i)$. An example is

$$A^+ = \left\{(\mu_r) : B_r^- \le \mu_r / \mu_{r+1} \le B_r^+, r = 1, \ldots, s-1\right\} \quad (1.3)$$

where B_r^-, B_r^+ represent fixed lower and upper bounds for these output multipliers. A^- applies similarly to input multipliers. (Note that (1.1) is referred to as an IDEA model as in Cooper, Park and Yu [7] when (1.1c) is replaced with $\mu, \omega \ge \varepsilon$. Here, ε represents a positive non-Archimedean element.) The AR bounds in the form of (1.3) are also intended to ensure positivity. Thus, throughout this chapter, we assume that all multipliers are to be positive.

1.2 Transformation to linear programming equivalents

We initiate our discussions of the transformations to be used by defining column maxima

y_r^o, x_i^o via

$$y_r^o = \max_j \left\{y_{rj}^o = \max[y_{rj} : (y_{rj}) \in D_r^+]\right\},\ r = 1, \ldots, s$$

$$x_i^o = \max_j \left\{x_{ij}^o = \max[x_{ij} : (x_{ij}) \in D_i^-]\right\},\quad i = 1, \ldots, m. \quad (1.4)$$

Thus, $y_r^o = \max\left\{ y_{r1}^o, ..., y_{rj}^o, ..., y_{rn}^o \right\}$ with components $y_{rj}^o = \max[y_{rj} : (y_{rj}) \in D_r^+],\ j = 1, ..., n$. Analogous treatments apply to the input data used to define x_i^o.

Such column maxima are readily identified for bounded data as in (1.2a). Trouble can be encountered, however, in the case of ordinal data as in (1.2b), where no maximum value is prescribed and such a column maximum may go to infinity. This would be possible in some of the semi-infinite programming formulations for DEA. See, for instance, Charnes, Cooper and Wei [6]. Here, however, we are confining our attention to $D_r^+ \subseteq L^n$, so we can assume a column maximum is present in *L*, a space of labels with an inherent ordinal ordering. Alternatively, we can choose an arbitrary positive real number for this column maximum without disturbing the indicated ordinal relations. In either case the problem associated with the need for having column maxima is eliminated from the transformations that we describe below.

To reduce the AR-IDEA model in (1.1) to an ordinary linear programming problem, we first rescale all data by means of the following formula:

$$\hat{y}_{rj} = y_{rj} / y_r^o,\ r = 1, ..., s$$
$$\hat{x}_{ij} = x_{ij} / x_i^o,\ \ i = 1, ..., m. \qquad (1.5)$$

Using these rescaled data in (1.1) we then obtain

$$\left.\begin{aligned} &\max \hat{\pi}_0 = \sum_{r=1}^{s} \hat{\pi}_r \hat{y}_{r0} \\ &\text{s.t.} \qquad \sum_{r=1}^{s} \hat{\pi}_r \hat{y}_{rj} - \sum_{i=1}^{m} \hat{\omega}_r \hat{x}_{ij} \le 0,\ j = 1, ..., n \\ &\qquad\qquad \sum_{i=1}^{m} \hat{\omega}_i \hat{x}_{i0} = 1 \end{aligned}\right\} \quad (1.1'\text{a})$$

$$\left.\begin{array}{l}\hat{y}_r = \left(\hat{y}_{rj}\right) \in \hat{D}_r^+ r = 1,\dots,s \\ \hat{x}_i = \left(\hat{x}_{ij}\right) \in \hat{D}_r^- i = 1,\dots,m\end{array}\right\} \quad (1.1'b)$$

$$\left.\begin{array}{l}\hat{\mu} = \left(\hat{\mu}_r\right) \in \hat{A}^+ \\ \hat{\omega} = \left(\hat{\omega}_i\right) \in \hat{A}^-\end{array}\right\} \quad (1.1'c)$$

where we have employed $\hat{\mu}$, $\hat{\omega}$ in place of the μ, ω in (1.1). The following theorem then establishes the relationships between (1.1) and (1.1′) that justify our use of (1.1′) in the subsequent developments.

Theorem 1.1. (i) The optimal objective values of both (1.1) and (1.1′) are equal, and (ii) $\hat{\pi}_r = \hat{\pi}_r y_r^{\rho} \forall r, \hat{\omega}_i = \hat{\omega}_i x_i^{\rho} \forall i.$

Proof. By virtue of the column rule of linear programming as given in Charnes and Cooper [3, p 29], it is clear that the optimal objective value of (1.1) is equal to the optimal objective in the modified model (1.1′). It therefore follows that $\hat{y}_{rj}\hat{\pi}_r = y_{rj}\pi_r$ and $\hat{x}_{ij}\hat{\omega}_i = x_{ij}\omega_i$ and using (1.5) completes this proof. For a statement and proof of the column rule of the column rule of linear programming is stated as interpreted in Charnes and Cooper [3] (p. 29), not proved by them, we here establish the following :

Column rule : The coefficients in any column of a linear programming problem (including the criterion elements) may all be multiplied or divided by a positive number without changing the optimal value of the objective.

Proof. Our proof is developed for the general linear programming problem which we formulate as follows:

$$\left.\begin{aligned} &\max = \sum_{j=1}^{n} c_j \lambda_j \\ &s.t \quad \sum_{j=1}^{n} a_{rj} \lambda_j \le b_i, i = 1,\ldots,m \\ &\qquad\qquad \lambda_j \ge 0,\ j = 1,\ldots,n \end{aligned}\right\} \qquad (1.6)$$

We assume that this problem has a finite optimum which we represent by $\lambda_j^* \ge 0,\ j = 1,\ldots,n$.

Now suppose that each of the columns is divided by a positive constant $d_j > 0$ which replaces (1.6) by

$$\left.\begin{aligned} &\max = \sum_{j=1}^{n} \hat{c}_j \lambda_j \\ &s.t \quad \sum_{j=1}^{n} \hat{a}_{rj} \lambda_j \le b_i, i = 1,\ldots,m \\ &\qquad\qquad \lambda_j \ge 0,\ j = 1,\ldots,n \end{aligned}\right\} \qquad (1.7)$$

where $\hat{c}_j = c_j / d_j$ and $\hat{a}_{ij} = a_{ij} / d_j$ for all i and j with $d_j = 1$ for those columns which are not to be changed.

Evidently $\hat{\lambda}_j = \lambda_j^* d_j,\ j = 1,\ldots,n$ is a solution to (1.6), which it reproduces. Moreover, this solution is optimal for (1.7). For suppose we could have a solution λ_j^p for which

$$\sum_{j=1}^{n}\hat{c}_j\lambda_j^p > \sum_{j=1}^{n}\hat{c}_j\hat{\lambda}_j = \sum_{j=1}^{n}c_j\lambda_j^* \qquad (1.8)$$

However $\frac{a_{ij}}{d_j}\lambda_j^p = a_{ij}\left(\frac{\lambda_j^p}{d_j}\right)$ and therefore $\sum_{j=1}^{n}\hat{a}_{ij}\lambda_j^p = \sum_{j=1}^{n}\hat{a}_{ij}\left(\frac{\lambda_j^p}{d_j}\right) \le b_i, i=1,\ldots,m$

, so the constraints of (1.6) would then be satisfied. The assumption (1.7) would therefore give a solution to (1.6) with $\sum_{j=1}^{n}\hat{c}_j\lambda_j^p = \sum_{j=1}^{n}c_j\left(\frac{\lambda_j^p}{d_j}\right) > \sum_{j=1}^{n}c_j\lambda_j^*$ and thereby contradict the optimality of the λ_j^* values for (1.6).

Remark. There is an associated row rule given in Charnes and Cooper [3] (p. 29) which states that multiplication or division of any row of a linear programming problem by a positive constant does not change the solution set. This row rule can be joined with the duality theory of linear programming to provide an alternative (simpler) proof. The direct route used in the above proof, however, is better suited for this chapter. For instance, the condition $\hat{a}_{ij}\hat{\lambda}_j = \hat{a}_{ij}\lambda_j^*$, as used in the above proof, specializes to the equations $\hat{y}_{rj}\hat{\mu}_r = y_{rj}\mu_r$ and $\hat{x}_{ij}\hat{\omega}_i = x_{ij}\omega_i$ given in the proof of Theorem 1.1.

We now use the index j_r, r = 1, …, s and j_{s+i}, i = 1, …, m to mean that DMU$_{jr}$ has only the column maximum at unity for the rescaled data column corresponding to output r, and $\text{DMU}_{j_{s+i}}$ has only the column maximum at unity for the rescaled data column corresponding to input i. In other words, the index j_r is identified as j such that $\min[\hat{y}_{rj} : (\hat{y}_{rj}) \in \hat{D}_r^+] = 1,\ j = 1,\ldots,n$ and j_{s+i} is identified as j such that $\min[\hat{x}_{rj} : (\hat{x}_{rj}) \in \hat{D}_r^-] = 1,\ j = 1,\ldots,n$. We hereafter refer to DMU$_{jr}$, $\text{DMU}_{j_{s+i}}$ as column maximum DMUs for each r and i. For brevity we symbolize these column maximum DMUs as CMDs.

As already noted, Cooper, Park and Yu [7] assumed that there exists at least one CMD for each r and i-viz, DMU$_{jr}$, $1 \le j_r \le n$ and $\text{DMU}_{j_{s+i}}$, $1 \le j_{s+i} \le n$, which means that we can specify j_r, j_{s+i} from among the j = 1, ..., n available DMUs. Proceeding on this assumption, the nonlinear version of the AR-IDEA model in (1.1′) can be transformed into a linear programming equivalent as follows.

New variables Y_{rj} and X_{ij} are introduced:

$$y_{rj} = \hat{y}_{rj}\hat{\mu}_r, \; r = 1,\ldots,s; \; j = 1,\ldots,n$$
$$x_{ij} = \hat{x}_{ij}\hat{\omega}_i, \quad i = 1,\ldots,m; \; j = 1,\ldots,n \tag{1.9}$$

and let

$$Y_r^o = \hat{\mu}_r \max{}_j \left\{ \hat{y}_{rj}^p = \min\left[\hat{y}_{rj} : \left(\hat{y}_{rj}\right) \in \hat{D}_r^+ \right] \right\} = Y_{rj}$$
$$X_r^o = \hat{\omega}_i \max{}_j \left\{ \hat{x}_{ij}^p = \min\left[\hat{x}_{ij} : \left(\hat{x}_{ij}\right) \in \hat{D}_r^- \right] \right\} = X_{ij_{s+i}} \tag{1.10}$$

for each r and i. The output variable y_{rj_r} on the right represents the CMD and, by assumption, the index j_r is specified by $j + 1$ in $\hat{y}_{rj+1}^p = \max\left\{ \hat{y}_{r1}^p,\ldots,\hat{y}_{rj}^p,\hat{y}_{rj+1}^p,\ldots,\hat{y}_{rn}^p \right\}$. The input variable $X_{ij_{s+i}}$ follows similarly.

Based on the above definitions, we obtain the following results: Lemma 1 below provides rules for the data transformations for treating (1.2). Lemma 1.2 is directed to the AR transformations for treating (1.3). Finally, Theorem 1.2 summarizes all transformations and transforms the nonlinear programming model given by (1.1′) into an ordinary linear programming problem. For the proofs of all the lemmas and theorem, see Cooper, Park and Yu. [7, 8]

Lemma 1.1 The sets of constraints on the data in (1.2a) and (1.2b) can be replaced with the following new sets:

bounded data : $B_r^+ = \left\{ \left(y_{rj}\right) : \hat{y}_{rj}^- y_{rj_r} \le y_{rj} \le \hat{y}_{rj}^+ y_{rj_r}, \forall j \ne j_r \right\}$ (1.11a)

ordinal data : $B_r^+ = \left\{ \left(y_{rj}\right) : y_{rj} \le y_{rj+1}, \; j = 1,\ldots,n-1 \right\}$ (3.11b)

where $\hat{y}_{rj}^-, \hat{y}_{rj}^+$ are the rescaled data, using (1.5) for the $\hat{y}_{rj}^-, \hat{y}_{rj}^+$ given in (1.2a).

Lemma 1.2 The set of constraints on multipliers in (1.3) can be converted into the following new set:

$$\hat{B}_r^+ = \left\{ \left(y_{rj} \right) : \beta_{rj}^- \frac{y_r^o}{y_{r+1}^o} y_{r+1, j_{r+1}} \le y_{rj_r} \le \beta_{rj}^+ \frac{y_r^o}{y_{r+1}^o} y_{r+1, j_{r+1}}, r = 1, \ldots, s-1 \right\}.$$

Theorem 1.2. Model (1.1) can be transformed into the following linear programming problem:

$$\left.\begin{aligned} &\max \hat{\pi}_0 = \sum_{r=1}^{s} y_{r0} \\ &\text{s.t.} \qquad \sum_{r=1}^{s} y_{rj} - \sum_{i=1}^{m} x_{ij} \le 0,\ j = 1, \ldots, n \\ &\qquad\qquad \sum_{i=1}^{m} x_{i0} = 1 \end{aligned}\right\} \qquad (1.12a)$$

$$\left.\begin{aligned} &\hat{y}_r = \left(\hat{y}_{rj} \right) \in \hat{D}_r^+ r = 1, \ldots, s \\ &\hat{x}_i = \left(\hat{x}_{ij} \right) \in \hat{D}_r^- i = 1, \ldots, m \end{aligned}\right\} \qquad (1.12b)$$

$$\left.\begin{aligned} &\hat{\mu} = \left(\hat{\mu}_r \right) \in \hat{A}^+ \\ &\hat{\omega} = \left(\hat{\omega}_i \right) \in \hat{A}^- \end{aligned}\right\} \qquad (1.12c)$$

where all variables x_{ij}; y_{rj} are non-negative.

Extension

An example

Before proceeding further, we provide a numerical example via Table 1.1 to make things more concrete. This table portrays four DMUs that produce two outputs using a single input. Here there exists a CMD in the column under 'Input' - viz, $\mathrm{DMU}_{j_{s+i}} = \mathrm{DMU}_{j_3}$. We can thus specify $x_{ij_{s+1}}$ for $i = 1$ by x_{11} and use Lemma 1.1 to obtain the set B_1^-, as in (1.12b), which is here accorded the following formulation:

$$B_1^- = \{(x_{11},...,x_{14}) : x_{11} \geq x_{12} \geq x_{13} \geq x_{14}\}.$$

Table 1.1 : Data matrix with imprecise data

DMU *J*	Input	Output	
	Ordinal x_{1j}	**Fixed-bound** y_{1j}	**Ratio-bound** y_{2j}
1	4	[9, 15]	[0.9, 1.2]
2	3	[6, 12]	1
3	2	[6, 9]	[0.6, 0.9]
4	1	[3, 6]	[0.3, 0.6]

Rankings (4 ≡ highest rank; … ≡ lowest rank), viz. $x_{21} \geq x_{22} \geq x_{23} \geq {}_{24}$.

Turning from inputs to outputs, we now note that no CMD exists in the last two columns. For instance, let us consider the output data column under 'Fixed-bound', and write

$$\hat{D}_1^+ = \{(\hat{y}_{11},...,\hat{y}_{14}) : 0.6 \leq \hat{y}_{11} \leq 1,...,0.2 \leq \hat{y}_{14} \leq 0.4\}$$

which is obtained by using (1.5) - viz, normalizing the original data on the column maximum, $y_1^0 = 15$, as in (1.4). In D_1^+, there exists no DMU that has only the value at unity such that $\hat{y}_{1j_1} = \hat{y}_1^0 = 1, 1 \leq j_1 \leq 4.$ This implies that we cannot identify an index j_1 among the existing four DMUs, j = 1, …, 4 for use as in (1.10). Hence we cannot use (1.11a) in Lemma 1.1 to obtain the linear programming equivalent of (1.13) in the form of (1.12).

To deal with this problem, we introduce a new dummy variable with its value at unity – viz., we introduce $\hat{y}_{15} = 1$ in D_1^+ and obtain

$$\hat{D}_1^{+} = \{(\hat{y}_{11},...,\hat{y}_{15}) : 0.6 \leq \hat{y}_{11} / \hat{y}_{15} \leq 1,...,0.2 \leq \hat{y}_{14} / \hat{y}_{15} \leq 0.4, \hat{y}_{15} = 1\}.$$

These data are rescaled relative to the unity assigned to the dummy variable. Thus the permissible values for $\{(\hat{y}_{11},...,\hat{y}_{15})$ in $\hat{D}_1^+$ and $\hat{D}_1^{+'}$ are identical. Moreover, we have the CMD in $\hat{D}_1^{+'}$ and can hence unambiguously identify j_1 = 5.

We can now use Lemma 1.1 - viz, using $y_{1j_1} = y_{15}$ in (1.11a). We therefore obtain the following set for B_1^+ as in (1.12b):

$$B_1^+ = \{(y_{11},..., y_{15}) : 0.6 \le y_{15} \le y_{11} \le y_{15},..., 0.2 y_{15} \le y_{14} \le 0.4 y_{15}\}.$$

It should be noted that the new variable $y_{15} (= \hat{y}_{11}\hat{\mu}_1 = \hat{\mu}_1)$ is positive so the constraints in B_1^+ preserve the original data structure. We also note that y_{15} is not related to a new DMU to be used in the constraints as in (1.12a) but is used only to represent the output multiplier as $y_{15} = \hat{\mu}_1$. Thus y_{15} can be used in transforming the AR bound A_1^+, if available, into the new set B_1^+ by using $y_{1j_1} = y_{15}$ in Lemma 1.2.

Turning to the data for the ratio bounds in the last column of Table 1.1, we first obtain the set of rescaled data, $\hat{D}_2^+$ by normalizing y_{2j} on the column maximum $y_2^0 = 1.2$. We then introduce a dummy $\hat{D}_1^{+'} = \{(\hat{y}_{21},..., \hat{y}_{25}) : 0.75 \le \hat{y}_{21} / \hat{y}_{25} \le 1, \hat{y}_{22} / \hat{y}_{25} = 0.83,$
$0.5 \le \hat{y}_{23} / \hat{y}_{25} \le 0.75, 0.25 \le \hat{y}_{24} / \hat{y}_{25} \le 0.5, \hat{y}_{25} = 1\}$. Next, using $y_{2j_2} = y_{25}$ in (3.8a) of Lemma 1.1, we obtain

$$B_1^+ = \{(y_{21},..., y_{25}) : 0.75 \le y_{25} \le y_{21} \le y_{25}, y_{22} = 0.83 y_{25},$$

$$0.5 y_{25} \le y_{23} \le 0.25 y_{25} \le y_{24} \le 0.5 y_{25}\}.$$

Generalization

We now generalize the above example. As can be seen in Table 1.1, a CMD is absent for the bounded data. We thus focus on treatments of the bounded data for our generalization. For simplicity, we deal only with output data variables because a comparable development applies for input data variables.

For this purpose, we recall (1.2a) to write $D_r^+ = \{(y_{r1},..., y_{rn}), y_{rj}^- \le y_{rj} \le y_{rj}^+, j = 1,..., n\}$. Using the column maximum, which is given simply by $y_r^0 = \max_j \{y_{rj}^+\}$, we rescale these original data via (1.5) to obtain

$$\hat{D}_r^+ = \{(\hat{y}_{r1},...,\hat{y}_{rj}^+), j = 1,...,n\} \qquad (1.13)$$

It is assumed that there does not exist DMU$_{jr}$, $1 \le jr \le$ n, that has only unity in this column r such that $\hat{y}_{rj_r} = 1$ to require $\hat{y}_{rj_r}^- = \hat{y}_{rj_r}^+ = 1$.

We introduce a dummy variable into (10) with its value at unity, $\hat{y}_{r,n+1} = 1$, and obtain

$$\hat{D}_r^+ = \{(\hat{y}_{r1},...,\hat{y}_{r,n+1}) : y_{rj}^- \le \hat{y}_{rj} / \hat{y}_{r,n+1} \le \hat{y}_{rj}^+, \quad \hat{y}_{r,n+1} = 1; j = 1,...,n\}. \qquad (1.10')$$

It is clear that the permissible values for the variables $(\hat{y}_{r1},...,\hat{y}_{rn})$ in $\hat{D}_r^+$ of (1.13) and $\hat{D}_r^{+'}$ of (1.13′) are identical. In (1.13′), moreover, we have $\hat{y}_{r1},...,\hat{y}_{r,n+1} = 1$ as a CMD in this column which is augmented by the dummy value at unity.

We can now use the methods presented in Cooper, Park and Yu [7] to achieve a linear programming equivalent as in (1.12). This is done with the following slight modifications. Let us define

$$y_{rj} = \hat{y}_{rj}\hat{\mu}_r, r = 1,...,s; j = 1,...,n+1 \qquad (1.14)$$

and, further,

$$y_r^0 = \hat{\mu}_r \max_{j=1,...,n+1}\left\{\hat{y}_{rj}^p = \min[\hat{y}_{rj} : (\hat{y}_{rj}) \in \hat{D}_r^{+1}]\right\} = y_{rj_r} \qquad (1.15)$$

for each $r = 1, \ldots, s$.

We then denote $\hat{y}_{rj}^p = \max_{j=1,...,n+1}\left\{\hat{y}_{rj}^p\right\} = \max\left\{\hat{y}_{rj}^p,...,\hat{y}_{r,n+1}^p\right\}$. It follows that $\hat{y}_r^0 = \hat{y}_{r,n+1}^p = 1$ and $y_{rj_r} = y_{r,n+1}$ represent the CMD. We thus have

$$y_r^0 = \hat{y}_r^0\hat{\mu}_r = y_{r,n+1}, r = 1,...,s \qquad (1.16)$$

and, further, by (1.14) and (1.16),

$$\hat{y}_{rj} = y_{rj} / y_{r,n+1}, r = 1,\ldots,s;\ j = 1,\ldots,n. \qquad (1.17)$$

To complete the transformation to a linear programming equivalent, we employ the following lemmas:

Lemma 1.3 The set $\hat{D}_r^+$ in (1.13) is replaced by

$$B_1^+ = \{(y_{r1},\ldots,y_{r,n+1}) : \hat{y}_{rj}^- y_{r,n+1} \le y_{rj} \le \hat{y}_{rj}^+ y_{r,n+1},\ j = 1,\ldots,n\}$$ **Proof.** This follows directly from (1.17) with $\hat{\mu}_r > 0$ and hence $y_{r,n+1} = 0$.

Lemma 1.4 Assume that column *r* + 1 has a maximum $\mathrm{DMU}_{j_{r+1}}$ while this is not the case for column *r*. Then, the set A^+ in (1.3) can be converted into

$$\hat{B}^+ = \left\{ \left(y_{rj_r} \right) : B_r^- \frac{y_r^0}{y_{r+1}^0} y_{r+1,j_{r+1}} \le y_{r,n+1} \le B_r^+ \frac{y_r^0}{y_{r+1}^0} y_{r+1,j_{r+1}}, r = 1,\ldots,s-1 \right\}.$$

Proof. This follows from Theorem 1.1(ii) and noting $\hat{\mu}_{r+1} = y_{r+1,j_{r+1}}$ and $\hat{\mu}_r = y_{r,n+1}$.

1.3 Conclusion and discussion

This gives us all we require to achieve the transformation to the linear programming equivalent given by (1.12) in the presence and/or the absence of CMDs. In particular, all of the sets $\hat{B}_r^+, B_i^-$ and $\hat{B}^+, \hat{B}^-$ that represent constraints in (1.12) are defined in Lemmas 1.1 and 1.2 for the case when CMDs are present and in Lemmas 1.3 and 1.4 for the case when they are absent.

A point we want to note is that the bounds on the multiplier variables, as in the AR approach, are translated into constraints on the variables associated with the data in the approach of Cooper, Park and Yu. [7] This implies that the incorporation of AR bounds into DEA affects the transformation of the data used in DEA (like those described in the cone-ratio envelopment of Charnes et al, [4] and also Charnes et al. [13] and Brockett et al. [2]). This same intent is continued into the present chapter and an extension is made such that the introduction of dummy variables with data enables us to transform multiplier variables to data variables when CMDs are absent in the imprecisely known data matrix.

However, there may be other ways besides introducing dummy variables, if we want to concentrate only on translating nonlinear versions of IDEA and AR-IDEA into linear programming equivalents. Thus, it is possible to do this by using only the employment of new variables as in (1.9).

To make things more concrete, we consider the following IDEA model in which the data are rescaled using (1.5):

$$\left.\begin{aligned} &\max \hat{\pi}_0 = \sum_{r=1}^{s} \hat{\mu}_r \hat{y}_{r0} \\ &s.t \qquad \sum_{r=1}^{s} \hat{\mu}_r \hat{y}_{rj} - \sum_{i=1}^{m} \hat{\omega}_i \hat{x}_{ij} \leq 0 \forall_j \\ &\qquad\qquad \sum_{i=1}^{m} \hat{\omega}_i \hat{x}_{i0} = 1 \end{aligned}\right\} \qquad (1.18a)$$

$$\left.\begin{aligned} &\hat{y}_r \in \hat{D}_r^+ = \{(\hat{y}_{rj}) : \hat{y}_{rj}^- \leq \hat{y}_{rj} \leq \hat{y}_{rj}^+, \forall_j\} \forall_r \\ &x_i \in \hat{D}_r^+ = \{(\hat{x}_{ij}) : \hat{x}_{ij} \leq \hat{x}_{ik}, \forall_j, k; j \neq k\} \forall_i \end{aligned}\right\} \qquad (1.18b)$$

$$\varepsilon \leq \hat{\mu}_r \hat{\omega}_i, \forall_r, i \qquad (1.18c)$$

Here, as shown in (1.18b), we assume that all output values lie within prescribed bounds and all input values are known to satisfy (weak) ordinal relations. The index *k* is used to represent DMUs, *k* = 1, …, *n*.

Now using (1.9) - viz, $y_{rj} = \hat{y}_{rj}\hat{\mu}_r, \hat{x}_{ij}\hat{\omega}_i, \forall_r, i, j$ - in (1.18), we have

$$\left.\begin{aligned} &\max \hat{\pi}_0 = \sum_{r=1}^{s} y_{r0} \\ &s.t \qquad \sum_{r=1}^{s} y_{rj} - \sum_{i=1}^{m} \hat{x}_{ij} \le 0, \forall_j \\ &\qquad\qquad \sum_{i=1}^{m} \hat{x}_{i0} = 1 \end{aligned}\right\} \quad (1.19a)$$

$$\left.\begin{aligned} &\hat{y}_r \in B_r^+ = \{(y_{rj}) : \hat{y}_{rj}^- \hat{\mu}_r \le y_{rj} \le \hat{y}_{rj}^+ \hat{\mu}_r, \forall_j\} \forall_r \\ &x_i \in B_r^- = \{(x_{ij}) : x_{ij} \le x_{ik}, \forall_j, k; j \ne k\} \forall_i \end{aligned}\right\} \quad (1.19b)$$

$$\varepsilon \le \hat{\mu}_r, \hat{\omega}_i, \forall_r, i \quad (1.19c)$$

Thus, we more simply achieve the linear programming equivalent to (1.18) without introducing dummy variables, as was done in the 'Extension' section, when CMDs are absent. (Note that this method can also be applied when CMDs are present.)

We know that the latter approach (i.e., moving from (1.18) to (1.19)) is simpler than our previous development for transforming the nonlinear problem (1.18) into the linear programming equivalent (1.19). There are, however, advantages to our approach. For instance, our approach enables us to directly see what relations obtain between multiplier variables and data variables. As an example, from (1.15) and (1.16), we know that $\hat{\omega}_i = \max_{j=1,\ldots n}\{(x_{ij} : x_{ij}) \in B_i^-\} = x_{ij\,s+i}$. Thus, (1.15) and (1.16) are needed to reduce the number of variables used in the linear programming problems if CMDs are present.

References

[1] Banker, R. D. (1993) : Maximum likelihood, consistency and data envelopment analysis: A statistical foundation. Mngmt. Sci., 39, p. 1265-1273.

[2] Brockett, P. L.; Charnes, A.; Cooper, W. W.; Huang, Z. M. and Sun, D.B. (1997) : Data transformations in DEA cone ratio envelopment approaches for monitoring bank performances. Eur. J. Oper. Res., 98, p. 250-268.

[3] Charnes, A. and Cooper, W. W. (1961) : Management Models and Industrial Applications of Linear Programming. John Wiley & Sons, New York.

[4] Charnes, A.; Cooper, W. W; Huang, Z. M. and Sun, D. B. (1990) : Polyhedral cone-ratio DEA models with an illustrative application to large commercial banks. J. Econom., 46, p. 73-91.

[5] Charnes, A.; Cooper, W. W.; Huang, Z. M. and Sun, D. B. (1991) : Relations between half-space and finitely generated cones in polyhedral cone-ratio DEA models. Int. J. Syst. Sci., 22, p. 2057-2077.

[6] Charnes, A.; Cooper, W. W. and Wei, Q. L. (1987) : A semi-infinite multicriteria programming approach to data envelopment analysis. Research Report CCS 551. Center for Cybernetic Studies, The University of Texas at Austin: Austin, TX.

[7] Cooper, W. W.; Park, K. S. and Yu, G. (1999) : IDEA and AR-IDEA: models for dealing with imprecise data in DEA. Mngmt. Sci., 45, p. 597-607.

[8] Cooper, W. W.; Park, K. S. and Yu, G. (2000) : An illustrative application of IDEA (Imprecise Data Envelopment Analysis) to a Korean mobile telecommunication company. Oper. Res.

[9] Olesen, O. B. and Petersen, N. C. (1995) : Chance constrained efficiency evaluation. Mngmt Sci. 41: 442-457.

[10] Olesen, O. B. and Petersen, N. C. (1999) : Probabilistic bounds on the virtual multipliers in data envelopment analysis: polyhedral cone constraints. J. Product Anal. 12, p. 103-134.

[11] Seiford, L. M and Zhu, J. (1998) : Sensitivity analysis of DEA models for simultaneous changes in all the data. J. Opl. Res. Soc., 49, p. 1060-1071.

[12] Thompson, R. G.; Dharmapala, P. S. and Thrall, R. M. (1995) : Linkedcone DEA profit rations and technical efficiency with application to Illinois coal mines. Int. J. Prod. Econ., 39, p. 99-115.

[13] Thompson, R. G.; Langemeier, L. N.; Lee, C. T. E. and Thrall, R. M. (1990) : The role of multiplier bounds in efficiency analysis with applications to Kansas farming. J. Econom., 46, p. 93-108.

*****