

## ALGORITHMIC STRATEGY FOR ASSURANCE REGION ANALYSIS IN DEA

Toshiyuki Sueyoshi  
*The Ohio State University*

(Received November 26, 1990; Revised March 12, 1991)

*Abstract* An algorithm strategy is proposed for use with the assurance region (AR) approach in data envelopment analysis (DEA). The strategy addressed in this study characterizes and classifies all decision making units (DMUs) into several subsets, using the revised simplex method of linear programming. Then, each DMU subset is solved by a different algorithm. Experimental studies consisting of randomly generated data sets have confirmed that the proposed algorithm outperforms the conventional DEA use of the revised simplex method. An important feature related to the DEA/AR algorithm is that it can deal effectively with large data sets.

### 1. Introduction

Charnes *et al.* [1] have opened up a new nonparametric approach, referred to data envelopment analysis (DEA), that can empirically determine the efficiency level of many organizations in public and privated sectors. The DEA development not only provides practitioners with an opportunity to enhance productive efficiency, but also provides researchers with numerous research issues related to efficiency measurement. As such example, Seiford [9] reports more than 400 DEA contributions in the last decade. In Japan, Tone [16] introduced the DEA technique and its related underlying concepts in terms of production economics.

In order to describe the research objective of this study, this article starts with describing two research issues on which many DEA researchers have been paying recent attention. First, a series of research works have investigated DEA algorithms and these related computation theory. The need for DEA algorithm development has been first argued by Charnes *et al.* [2, 3] because DEA applications require tremendous computation efforts. An analytical way of classifying DMUs is also proposed in [2, 3] so as to improve DEA algorithmic efficiency. Although [2, 3] did not present any computational results concerning the proposed algorithm, these studies described explicitly the importance and need for further algorithmic development. The contributions of [2, 3] are summarized in the forms of representation and classification theorems. Following the theorems; Chang and Sueyoshi [5], Sueyoshi and Chang [13], and Sueyoshi [10, 12] have presented efficient algorithms applied to different DEA models, incorporating several different algorithmic strategies.

The other important research area is to develop a new DEA approach (often referred to as "post-DEA approach") that is designed to measure the efficiency of each decision making unit (DMU) by restricting dual variables of DEA to acceptable ranges. The restriction on dual variables in the DEA method is important because it can incorporate *a priori* information concerning DEA efficiency measurement. As a consequence of such restriction, the following new perspectives are added to DEA applications:

- a. First, the restriction of dual variables reduces the number of efficient DMUs and

more sharply delineates the best DMU(s) for a decision maker. [See [11] for such an example representing how the number of efficient DMUs is reduced.]

- b. Second, the restriction provides DEA with an analytical scheme of measuring allocative efficiency (AE). [The conventional use of DEA focuses upon only technical efficiency (TE). The concept indicates the achievement of an efficient frontier. Meanwhile, the AE denotes the measure as to how a DMU operates, using input (output) quantities along the vector indicating by given input (output) prices. See [11] for a graphical description regarding the difference between TE and AE.]

In the DEA literature two approaches are proposed to restrict dual variables to previously specified ranges. That is, DEA/assurance region (DEA/AR) analysis is proposed in [14] and cone-ratio DEA method is proposed in [4]. [A detailed description concerning theoretical differences between the two post DEA approaches lies beyond the scope of this study.]

The purpose of this study is to develop an efficient algorithm for measuring TE and AE within the framework of the DEA/AR approach. In this sense, this study may be considered as an important extension of previous studies [5, 10, 11, 12, 13]. Furthermore, this research will document several results of experimental simulation studies applied to the newly developed DEA algorithm. The remainder of this article is organized as follows. Section 2 describes DEA/AR approach and describes its unique features in terms of algorithmic development. Section 3 states the DEA algorithm for AR that incorporates several computational strategies to enhance its algorithmic efficiency. Section 4 exhibits the results of a simulation study in which the proposed algorithm is compared with the conventional use of the revised simplex method for DEA/AR. Conclusion and future extensions are summarized in the last section.

## 2. DEA/AR Approach

### 2.1 DEA ratio form

This section reviews the original DEA ratio form and then extends its discussion into DEA/AR approach. Suppose that there are  $n$  DMUs denoted  $j \in J$ , each of which yields a nonzero output vector  $Y_j = (y_{1j}, y_{2j}, \dots, y_{sj})^T \geq 0$ , using a nonzero input vector  $X_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T \geq 0$ . Here, the term “nonzero” indicates that at least one vector component is positive. The superscript “ $T$ ” indicates the transpose of a vector. It is also assumed that there is no DMUs in  $J$  whose data domain is proportional to that of another DMU. [See Charnes *et al.* [3] for a discussion of this assumption.] Admitting that there are many different DEA models as presented in [9], this study focuses upon only the original DEA ratio form in which a specific DMU <sub>$o$</sub>  uses  $X_o$  to produce  $Y_o$ :

$$\begin{aligned}
 & \text{minimize } \theta \\
 & \text{subject to } - \sum_{j=1}^n x_{ij} \lambda_j + \theta x_{io} \geq 0, \quad i = 1, \dots, m \\
 & \quad \sum_{j=1}^n y_{rj} \lambda_j \geq y_{ro}, \quad r = 1, \dots, s, \\
 & \quad \lambda_j \geq 0, \quad j = 1, \dots, n.
 \end{aligned} \tag{1}$$

As shown in [1], (1) is a linear programming form equivalent to the original nonlinear ratio form. The optimal  $\theta$  indicates a DEA efficiency score on the closed interval  $[0,1]$ . The vector  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)^T$  is used to construct a convex hull covering all data points.

The dual form of (1) becomes

$$\begin{aligned}
 &\text{maximize } \eta = \sum_{r=1}^s u_r y_{ro} \\
 &\text{subject to } - \sum_{i=1}^m v_i x_{ij} + \sum_{r=1}^s u_r y_{rj} \leq 0, \quad j = 1, \dots, n \\
 &\quad \sum_{i=1}^m v_i x_{io} = 1, \\
 &\quad v_i \geq 0, i = 1, \dots, m \text{ and } u_r \geq 0, r = 1, \dots, s,
 \end{aligned} \tag{2}$$

where  $v_i$  and  $u_r$  are dual variables related to input and output constraints of (1). At optimality  $\theta = \eta$  is always achieved and both (1) and (2) produce the same DEA efficiency score indicating TE.

As described in [10,11], there are several important features related to (1).

- a. First, (1) is used  $n$  times in a computational process that changes the values of  $X_o$  and  $Y_o$  according to which DMU is being measured, while maintaining the same values of  $X_j$  and  $Y_j (j = 1, \dots, n)$ .
- b. Second,  $n$  (the number of DMUs) is usually much larger than  $m + s$  (the number of inputs plus outputs) in DEA data sets. Therefore, the number of columns (representing  $\theta$  and  $\lambda_j, j = 1, \dots, n$ ) dominates the number of rows (for inputs and outputs) in the simplex tableau of (1).

### 2.2 DEA/AR model

The DEA ratio form, as noted previously, proceeds entirely from an observational data set. In this sense, it is “value free.” However, there are many cases where it may be desirable to incorporate information that is known to provide meaningful restrictions on *a priori* grounds. The recent work by Thompson *et al.* [14] makes it possible to achieve such a restriction by means of AR. [An alternative approach referred to as “cone-ratio approach” is discussed in [4].]

Following [14], the DEA/AR can be formally defined by

$$\begin{aligned}
 \alpha_i^L v_1 &\leq v_i \leq \alpha_i^U v_1, i = 2, \dots, m \\
 \beta_r^L u_1 &\leq u_r \leq \beta_r^U u_1, r = 2, \dots, s,
 \end{aligned} \tag{3}$$

where the positive scalars  $\alpha_i^L$  and  $\alpha_i^U$  represent prescribed lower and upper bounds for the values of the ratio of dual variables  $v_i/v_1$  for each  $i = 2, \dots, m$  and  $u_r/u_1$  for each  $r = 2, \dots, s$ . [A detailed discussion AR and its underlying theory can be found in [14, 15].]

The DEA/AR used in this article can thus be formally expressed by appending (3) to (2). The resulting DEA/AR form becomes

$$\begin{aligned}
 &\text{maximize } \sum_{r=1}^s u_r y_{ro} \tag{4} \\
 &\text{subject to } \sum_{i=1}^m v_i x_{ij} + \sum_{r=1}^s u_r y_{rj} \leq 0, \quad j = 1, \dots, n, \tag{4-1} \\
 &\quad \sum_{i=1}^m v_i x_{io} = 1, \tag{4-2}
 \end{aligned}$$

$$v_i - \alpha_i^U v_1 \leq 0, \quad i = 2, \dots, m, \quad (4-3)$$

$$-v_i + \alpha_i^L v_1 \leq 0, \quad i = 2, \dots, m, \quad (4-4)$$

$$u_r - \beta_r^U u_1 \leq 0, \quad r = 2, \dots, s, \quad (4-5)$$

$$-u_r + \beta_r^L u_1 \leq 0, \quad r = 2, \dots, s, \quad (4-6)$$

$$v_i \geq 0, i = 1, \dots, m \text{ and } u_r \geq 0, \quad r = 1, \dots, s.$$

The primal form corresponding to (4) can be formulated as

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } -\sum_{j=1}^n x_{1j} \lambda_j + x_{1o} \theta - \sum_{i=2}^m \alpha_i^U \eta_i^U + \sum_{i=2}^m \alpha_i^L \eta_i^L \geq 0, \\ & \quad -\sum_{j=1}^n x_{ij} \lambda_j + x_{io} \theta + \eta_i^U - \eta_i^L \geq 0, \quad i = 2, \dots, m, \\ & \quad \sum_{j=1}^n y_{1j} \lambda_j - \sum_{r=2}^s \beta_r^U \delta_r^U + \sum_{r=2}^s \beta_r^L \delta_r^L \geq y_{1o}, \\ & \quad \sum_{j=1}^n y_{rj} \lambda_j + \delta_r^U - \delta_r^L \geq y_{ro}, \quad r = 2, \dots, s, \\ & \quad \lambda_j \geq 0, j = 1, \dots, n, \eta_i^U \geq 0, \eta_i^L \geq 0, \quad i = 2, \dots, m, \text{ and} \\ & \quad \delta_r^U \geq 0, \delta_r^L \geq 0, \quad r = 2, \dots, s, \end{aligned} \quad (5)$$

where  $\eta_i^U$  and  $\eta_i^L, i = 2, \dots, m$  are decision variables derived from (4-3) and (4-4), respectively. Also,  $\delta_r^U$  and  $\delta_r^L, r = 2, \dots, s$  are decision variables associated with (4-5) and (4-6), respectively. Finally,  $\theta$  continues to be associated with (4-2), which is the same form as presented in (1). In (5) this study maintains the assumption that the number of DMUs dominates that of inputs plus outputs. Hence, the number of columns is much larger than that of rows in the simplex tableau of (5).

### 2.3 Characterization of DMUs

The DEA algorithm proposed for AR approach, where the number of inputs plus outputs is dominated by that of DMUs, characterizes the type of DMUs and thereby partitions all DMUs into several subsets. Each subset is solved by a different computation process that takes advantage of the two algorithmic properties described before.

Using the result of [3], this study classifies DMUs into the following subset:

$$J = E \cup E' \cup F \cup N. \quad (6)$$

Here,  $J$  is the whole set of all DMUs. Each subset can be defined in the following matter:

- a.  $E$  is the set of technically efficient DMUs. Each DMU in  $E$  has a unique primal optimal solution with  $\theta^* = 1, \lambda_o^* = 1, \lambda_j^* = 0$  for all  $j \neq o \in J$ , and all zero slack variables in (1). [See [11] for the definitions of slack variable in (1).]
- b.  $E'$  is the set of technically efficient DMUs as well. Each DMU in  $E'$  has alternate optimal solutions with  $\theta^* = 1, \lambda_o^* < 1, \lambda_j^* > 0$  for some  $j \neq o \in E$  and all zero slack variables in (1).
- c.  $F$  is the set of DMUs that have  $\theta^* = 1$  but are technically inefficient because these DMUs have at least one positive slack variable(s) in (1).

- d.  $N$  is the set of inefficient DMUs. Each DMU has a unique optimal  $\lambda_j^* > 0$  for some  $j \neq o \in E$  and at least one positive slack variable(s) in (1).

[Since a graphical description regarding the four subsets can be found in [11], this study avoids duplicating the same figure here. See Figure 1 of [11].]

This study classifies (6) further in a way that it can be more easily incorporated in algorithmic development for DEA as discovered in [10, 12]. The subsets decomposed here are  $F$  and  $N$  and the two subsets become  $F = IF \cup IF'$  and  $N = IE \cup IE'$ . Thereby,  $J$  can be decomposed as

$$J = E \cup E' \cup IF \cup IF' \cup IE \cup IE'. \quad (7)$$

An important feature related to (7), that distinguishes itself from (6), is the use of the concept of dominance in terms of Pareto optimality. [See [8] for the concept. The concept of Pareto optimality has served as a cornerstone of neoclassical economic theory and modern multi-criteria decision making theory.] This study utilizes the concept for algorithmic development (not for theoretical extension) so as to increase the computational efficiency of DEA applications. In order to describe the use of the concept in terms of DEA algorithmic development, it is convenient to express the concept using a simple vector comparison. Suppose that the  $j$ th DMU uses an input vector  $X_j = (x_{1j}, x_{2j}, \dots, y_{mj})^T$  to yield an output vector  $Y_j = (y_{1j}, y_{2j}, \dots, y_{sj})^T$ . If the following condition:

$$[x_{1j}, \dots, x_{mj}, -y_{1j}, \dots, -y_{sj}]^T \geq k[x'_{1j}, \dots, x'_{mj}, -y'_{1j}, \dots, -y'_{sj}]^T \quad (8)$$

is satisfied, then the  $j$ th DMU is dominated by the  $j'$ th DMU. Here, inequality must hold for at least one component of (8). Furthermore,  $k = \min x_{ij}/x'_{ij}$ , assuming that  $x'_{ij} \neq 0$ . [See [3] for a detailed discussion regarding the value of  $k$ .]

The concept of dominance partitions the set  $J$  into its two subsets  $J = J_n \cup J_d$ , where

$$J_d = \{j \mid \text{there exists some } j' \text{ satisfies (8)}\} \text{ and } J_n = J - J_d. \quad (9)$$

The set  $J_d$  is referred to as a “dominated set” and  $J_n$  as a “nondominated set.” The concept of dominance separates  $N$  and  $F$  into four subsets as follows:

$$N = IE \cup IE' \text{ and } F = IF \cup IF',$$

where  $IE = N \cap J_d$ ,  $IE' = N - IE$ ,  $IF = F \cap J_d$  and  $IF' = F - IF$ . The symbol “ $\cap$ ” indicates the intersection of two sets. It is obvious that  $E \cup E'$  belongs to  $J_n$ . Consequently,  $J$  can be decomposed by

$$J = J_d \cup J_n, \quad (10)$$

where  $J_d = IF \cup IE$  and  $J_n = E \cup E' \cup IF' \cup IE'$ . It is important to note that there is another way of classifying  $J$  as presented in [2]. The classification of (10) incorporates the concept of dominance, while [2] does not include it for the classification of  $J$ .

The characterization and classification of DMUs discussed above is useful for increasing algorithmic efficiency of (1). In order to extend the algorithm of (1) into DEA/AR (4) that needs to restrict dual variables, this study proposes a new type of DMU set in addition to the six subsets ( $J = E \cup E' \cup IE \cup IE' \cup IF \cup IF'$ ). Let us denote the set by EP that is formally defined as

- e. EP is the set of DMUs in  $E \cup E'$  whose sets of optimal dual variables measured by (1) satisfy AR condition (3). Each DMU in EP satisfies  $\theta^* = 1$  at optimal DEA/AR (4).

### 3. Algorithm

The algorithm proposed for DEA/AR has the four major computation processes as follows:

- a. First, as a preparatory process, the algorithm decomposes  $J$  into  $J_n$  and  $J_d$  by (8). Suppose that there are  $n$  DMUs in  $J$  that are partitioned into  $n_1$  DMUs in  $J_n$  and  $n_2$  DMUs in  $J_d$ . [Of course,  $n = n_1 + n_2$ .] The preparatory process temporarily excludes the columns of  $\lambda_j$  related to explicitly inefficient DMUs in  $J_d$  from the simplex tableau of (1). [Note that DMUs in  $J_d$  are examined after the TE measurement of all DMUs in  $J_n$ .]
- b. Second, using (1), the algorithm measures TE of each DMU in  $J_n$  and gradually suppresses the columns of  $\lambda_j$  related to DMUs in  $E' \cup IE' \cup IF'$  from its simplex tableau. As a consequence of this stage, the DEA algorithm for measuring TE can classify  $J_n$  into the four different subsets (i.e.,  $J_n = E \cup E' \cup IE' \cup IF'$ ).
- c. Third, using (1), the algorithm proceeds to measure TE of DMUs in  $J_d$ , maintaining only  $\lambda_j$  columns related to DMUs  $j \in E$  in the tableau. Let  $n_e$  be the number of DMUs belonging to  $E$ . The number of  $\lambda_j$  columns is reduced from  $n$  (for  $J$ ) to  $n_e$  (for  $E$ ). Thereby, the DEA algorithm can increase its computational efficiency to measure TE of DMUs in  $J_d$ .
- d. Finally, the last stage of the DEA algorithm measures AE of all DMUs, using (5). The important algorithmic aspect of the computation process is that (5) maintains only  $\lambda_j$  columns,  $j \in E \cup E'$ , in the simplex tableau. Thus, the reduced column size can reduce considerably the computation time and effort for measuring AE of all the DMUs in  $J$ .

#### 3.1 Reordered DMU set

In order to determine an incoming DMU to be measured, the DEA algorithm reorders DMUs in  $J$  according to the values of

$$g_j = y_j/x_j, \quad j \in J \quad (11)$$

where  $x_j = \sum_i x_{ij}$  and  $y_j = \sum_r y_{rj}$ . The algorithm uses a descending order of the  $g_j$  values to select an incoming DMU when classifying  $J$  into  $J_n$  and  $J_d$ . Meanwhile, it uses an ascending order of the  $g_j$  value when selecting an incoming DMU to measure its TE and AE.

Henceforth, this study uses a new symbol “ $\rho$ ” to represent the computational order of DMUs to be explored by the DEA algorithm. The algorithmic order is referred to as a “stage.” It is important to distinguish the difference between  $j$  and  $\rho$ . The symbol  $j$  indicates the observed order of DMUs, while the  $\rho$  denotes the algorithmic order of DMUs to be measured. Hence,  $\rho = 1$  denotes the initial stage of the algorithm and  $j = 1$  indicates the first observed DMU.

#### 3.2 Algorithm for TE measurement of $J_n$

After classifying  $J = J_n \cup J_d$  by (8), the DEA algorithm temporarily excludes the explicitly inefficient DMUs in  $J_d$  from its computational process and then initiates itself by setting

$$B_1 = J_n, C_1 = \phi, E_1 = \phi \text{ and } E'_1 = \phi,$$

where  $B, C, E$  and  $E'$  indicate four different subsets of  $J_n$  that are used for algorithmic convenience only. The symbol “ $\phi$ ” denotes an empty set. The subscript “1” is used to

represent the initial stage of the four subsets. Thus, the four sets at the  $\rho$ th stage can be expressed by  $B_\rho, C_\rho, E_\rho$ , and  $E'_\rho (\rho = 1, \dots, n)$ . The four sets can be formally defined as

- a.  $E_\rho = \{\text{DMU}_j \mid j \in E \text{ and the } j\text{th DMU has been examined}\}$ ,
- b.  $E'_\rho = \{\text{DMU}_j \mid j \in E' \text{ and the } j\text{th DMU has been examined}\}$ ,
- c.  $B_\rho = \{\text{DMU}_j \mid j \in E_\rho\} \cup \{\text{DMU}_j \mid \text{the } j\text{th DMU in } J_n \text{ has not yet been examined}\}$ ,
- d.  $C_\rho = \{\text{DMU}_j \mid \lambda_j^* = 1 \text{ and the } j\text{th DMU in } J_n \text{ has not yet been examined}\}$ .

Here,  $E_\rho$  and  $E'_\rho$  are two subsets of  $E$  and  $E'$  at the  $\rho$ th stage, respectively.  $B_\rho$  is a union set of  $E_\rho$  and the DMU set in  $J_n$  whose DEA measurement is not yet examined.  $C_\rho$  is a set of DMUs whose optimal  $\lambda_j^*$  value is observed as one at the stage  $\rho$ .

Now, in order to describe the use of the four subsets for algorithmic development, suppose that the algorithm is currently solving the  $\text{DMU}_\rho$  after setting  $\rho = j, j \in B_\rho - E_\rho$ . Here,  $B_\rho - E_\rho$  indicates a DMU set that contains all the unexamined DMUs in  $J_n$  at the  $\rho$ th stage. Then, the algorithm solves the TE measurement of the  $\text{DMU}_\rho$  by the following DEA model:

$$\begin{aligned}
 & \text{minimize } \theta \\
 & \text{subject to } - \sum_{j \in B_\rho} x_{ij} \lambda_j + \theta x_{i\rho} \geq 0, \quad i = 1, \dots, m, \\
 & \quad \sum_{j \in B_\rho} y_{rj} \lambda_j \geq y_{r\rho}, \quad r = 1, \dots, s, \\
 & \quad \lambda_j \geq 0, j \in B_\rho.
 \end{aligned} \tag{12}$$

The difference between (1) and (12) is that (1) uses all  $\lambda_j, j \in J$  as decision variables, while (12) uses its subset  $B_\rho (\subseteq J)$  so as to measure DEA solutions of the  $\text{DMU}_\rho$ . As an algorithmic advantage, the size of  $B_\rho$  is always diminishing as the algorithm proceeds.

Based upon DEA solutions related to (12), the  $\text{DMU}_\rho$  is classified by the following way:

- a. If  $\theta^* < 1$ , then the  $\text{DMU}_\rho$  belongs to  $IE'$ . Set  $B_{\rho+1} = B_\rho - \{\text{DMU}_\rho\}$ ,  $E_{\rho+1} = E_\rho$ ,  $E'_{\rho+1} = E'_\rho$  and  $C_{\rho+1} = \{\text{DMU}_j \mid \lambda_j^* > 0 \text{ and the DMU}_j \text{ has not yet been examined}\}$ .
- b. If  $\theta^* = 1, \lambda_\rho^* = 1, \lambda_j^* = 0$  for all  $j \neq \rho$  and all slack variables are zero, then the  $\text{DMU}_\rho$  belongs to  $E$ . Set  $B_{\rho+1} = B_\rho$ ,  $E_{\rho+1} = E_\rho \cup \{\text{DMU}_\rho\}$ ,  $E'_{\rho+1} = E'_\rho$  and  $C_{\rho+1} = C_\rho$ .
- c. If  $\theta^* = 1, \lambda_\rho^* < 1, \lambda_j^* > 0$  for some  $j \neq \rho$  and all slack variables are zero, then the  $\text{DMU}_\rho$  belongs to  $E'$ . Set  $B_{\rho+1} = B_\rho - \{\text{DMU}_\rho\}$ ,  $E_{\rho+1} = E_\rho$ ,  $E'_{\rho+1} = E'_\rho \cup \{\text{DMU}_\rho\}$  and  $C_{\rho+1} = \{\text{DMU}_j \mid \lambda_j^* > 0 \text{ and the DMU}_j \text{ has not yet been examined}\}$ .
- d. If  $\theta^* = 1$  and at least one slack variable is positive, then the  $\text{DMU}_\rho$  belongs to  $IF'$ . Set  $B_{\rho+1} = B_\rho - \{\text{DMU}_\rho\}$ ,  $E_{\rho+1} = E_\rho$ ,  $E'_{\rho+1} = E'_\rho$  and  $C_{\rho+1} = \{\text{DMU}_j \mid \lambda_j^* > 0 \text{ and the DMU}_j \text{ has yet been examined}\}$ .

If  $C_\rho$  is not an empty set, the DEA algorithm selects an incoming  $\text{DMU}_\rho$  from  $C_\rho$  and starts with  $\lambda_\rho = 1$  as an initial entering variable of the simplex tableau of (12). The strategy is due to the fact that all the DMUs in  $C_\rho$  belongs to  $E \cup E'$ .

### 3.3 Algorithm for TE measurement of $J_d$

After measuring TE of all the DMUs in  $J_n$ , the algorithm sets  $\rho = j, j \in J_d$  to evaluate TE of the  $\text{DMU}_\rho$ , using the following DEA model:

$$\text{minimize } \theta$$

$$\begin{aligned}
 \text{subject to } & - \sum_{j \in E} x_{ij} \lambda_j + \theta x_{i\rho} \geq 0, \quad i = 1, \dots, m, \\
 & \sum_{j \in E} y_{rj} \lambda_j \geq y_{r\rho}, \quad r = 1, \dots, s, \\
 & \lambda_j \geq 0, \quad j \in E,
 \end{aligned} \tag{13}$$

The difference between (1) and (13) is that (1) uses the columns related to  $\lambda_j, j \in J$ , while (13) uses only those related to  $\lambda_j, j \in E$ . Thus, the number of  $\lambda_j$  columns is reduced from  $n$  to  $n_e$  in the simplex tableau. The reduced columns can improve algorithmic efficiency to measure TE of  $J_d$ , particularly when the size of  $J_d$  is large. The end of the stage indicates the completion of TE measurement regarding all DMUs in  $J$ .

### 3.4 Algorithm for AE measurement of $J$

The final stage of the DEA algorithm takes advantage of the computational results explored before. That is, measuring TE of DMUs in  $J$ , the algorithm classifies  $J$  into the six different subsets ( $J = E \cup E'IE \cup IE' \cup IF \cup IF'$ ). When developing the simplex tableau for AE measurement, the algorithm uses the  $\lambda_j$  columns related to only DMUs in  $E \cup E'$ , because the DMUs satisfying AE is a subset of  $E \cup E'$ . The algorithm needs to pay attention only on  $E \cup E'$ .

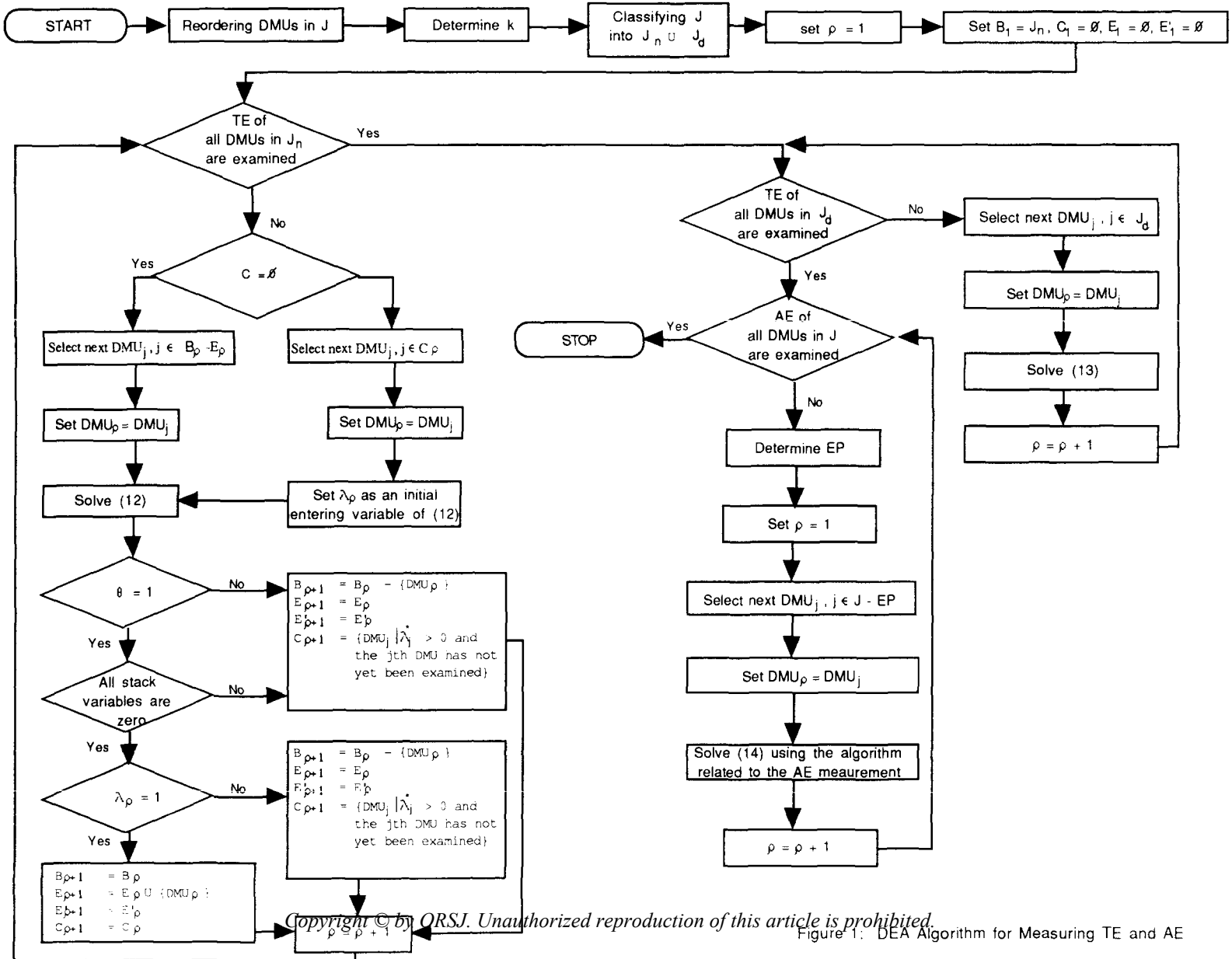
Now, in order to measure AE of DMUs in  $J$ , the DEA algorithm examines first whether dual variables of DMUs in  $E \cup E'$  satisfy AR requirement (3). If the dual variables satisfy (3), then the DMU can be identified to belong to EP and its AE measurement does not require any additional computation. Conversely, if the dual variables do not satisfy (3), then the AE of the DMU needs to be measured. The AE measurement of  $DMU_\rho$  violating (3) can be measured by the following DEA/AR model:

$$\begin{aligned}
 & \text{minimize } \theta \\
 \text{subject to } & - \sum_{j \in EP} x_{1j} \lambda_j + x_{1\rho} \theta - \sum_{i=2}^m \alpha_i^U \eta_i^U + \sum_{i=2}^m \alpha_i^L \eta_i^L \geq 0, \\
 & - \sum_{j \in EP} x_{ij} \lambda_j + x_{i\rho} \theta + \eta_i^U - \eta_i^L \geq 0, \quad i = 2, \dots, m, \\
 & \sum_{j \in EP} y_{rj} \lambda_j - \sum_{r=2}^s \beta_r^U \delta_r^U + \sum_{r=2}^s \beta_r^L \delta_r^L \geq y_{1\rho}, \\
 & \sum_{j \in EP} y_{rj} \lambda_j + \delta_r^U - \delta_r^L \geq y_{r\rho}, \quad r = 2, \dots, s \\
 & \lambda_j \geq 0, j = 1, \dots, n, \eta_i^U \geq 0, \eta_i^L \geq 0, i = 2, \dots, m \\
 & \text{and } \delta_r^U \geq 0, \delta_r^L \geq 0, r = 2, \dots, s.
 \end{aligned} \tag{14}$$

Here, each  $DMU_\rho$  is selected from  $J - EP$ . The difference between (5) and (14) is that (5) uses the  $\lambda_j$  columns related to all the DMUs in  $J$  in its simplex tableau, while (14) uses those for DMUs in  $EP$ . Since  $EP$  is a small subset of  $J$  in a real data set, the size of columns in the tableau of (14) becomes much smaller than that of (5). Consequently, (14) can be solved with reduced CPU time and algorithmic effort. The completion of the AR measurement indicates the end of the entire DEA algorithm for measuring TE and AE.

Finally, this article provides a step-by-step description concerning the whole algorithm visually summarized in Figure 1.





Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Figure 1: DEA Algorithm for Measuring TE and AE

## 4. Computational Results

### 4.1 Simulation study for TE measurement

Experimental simulation studies consisting of randomly generated data sets were carried out to compare the performance of two DEA codes related to TE measurement. The first coded (named “TE-revised simplex”) utilized the ordinary revised simplex method for (1) that did not incorporate any computational strategy. The code was a “dumb code” that always started from scratch to measure TE of each DMU according to its observed order. The other code (named “TE-special code”) used the DEA algorithm presented in Sections 3.1, 3.2 and 3.3. Thus, the object of this section is to examine the performance as to how the TE-special code improves the TE-revised simplex in terms of its algorithmic efficiency.

All the simulations were run on an IBM-AR. The total time spent in solving generated data sets were recorded in central processor unit (CPU) seconds. The structure of the simulation study was a  $2 \times 9 \times 5$  factorial experiment in which each treatment had 10 replications. Factors of the simulations are summarized as follows:

- a. Method (2 levels): TE-revised simplex and TE special code,
- b. Number of DMUs (9 levels): 20, 40, 60, 80, 100, 200, 300, 400, and 500,
- c. Percentage of efficient DMUs (5 levels): 10, 20, 30, 40 and 50(%).

This study generated six different DMU subsets (i.e.,  $E, E', IE, IE', IF$  and  $IF'$ ), following the data generation process proposed by Sueyoshi [10]. Each DMU uses two inputs to yield three outputs. This article avoids the duplication of describing the four DMU subsets ( $E, E', IE$  and  $IE'$ ), because the data generation process of the subsets is mentioned in detail by Sueyoshi [10]. Hence, this study focuses upon a description regarding how to generate DMUs in  $IF$  and  $IF'$ . It is important to describe here that all the DMU set ( $J$ ) is classified into  $J = E \cup E' \cup IE \cup IE'$  by the DEA-additive model in [10], while  $J$  is decomposed into  $J = E \cup E' \cup IE \cup IE' \cup IF \cup IF'$  by DEA-ratio form (1). Thus, the difference between the two DEA models can be characterized by the existence of such DMUs in  $IF \cup IF'$ . The rationale is due to the fact that the additive model is formulated on the concept of Farrell-Debreau efficiency analysis [6], while the ratio form is based upon the concept of Koopmans-Pareto production analysis [7].

The data generation of DMUs in  $IF \cup IF'$  can be described as follows: First, the simulation process selected a single DMU in  $E$  and then increased the quantity of one input measure, while maintaining the quantities of the other input and output measures. The resulting DMU belongs to  $IF$ . Thus, this study could generate two DMUs from each DMU in  $E$ . In order to describe how much the input quantity is increased, let the input vector of the DMU in  $E$  be  $(x_1, x_2, x_3)$ . Then, the DMU generates two input vectors denoted by  $(x'_1, x_2, x_3)$  and  $(x_1, x'_2, x_3)$  where  $x'_1 > 7/3x_2$  and  $x'_2 > 7/3x_1$ , respectively. [See [10, 13] for the rationale regarding the selection of 7/3.] The simulation process was repeated by selecting different DMUs in  $E$  to obtain many DMUs in  $IF$ . Second, in order to obtain DMUs in  $IF'$ , this study applied a convex combination to two adjacent DMUs in  $E$ . Then, one output quantity of the generated DMU was reduced, while the other output and inputs were maintained at these current levels. The simulation study was repeated many times by selecting different adjacent DMUs in  $E$  until it provided enough DMUs in  $IF'$ . Table 1

Table 1: Comparison between two DEA codes for measuring TE (average CPU seconds)

Efficient DMUs (%)	Inefficient DMUs (%)		Numbers of DMUs								
			20	40	60	80	100	200	300	400	500
E U E'	IE U IE' U IF U IF'										
10%	90%	TE-Revised Simplex	1.635	2.743	4.615	6.351	8.573	18.525	28.094	40.640	54.824
(9%)[1%]	(40%)[40%](5%)[5%]	TE-Special Code	.715	.910	1.191	1.525	1.861	3.110	4.260	5.930	7.784
20%	80%	TE-Revised Simplex	1.730	3.585	5.628	7.566	9.850	23.510	32.137	48.343	60.094
(18%)[2%]	(36%)[36%](4%)[4%]	TE-Special Code	.805	1.362	1.840	2.248	2.801	5.480	7.388	10.307	12.362
30%	70%	TE-Revised Simplex	2.015	4.180	6.000	8.242	10.796	23.960	35.817	48.975	63.222
(27%)[3%]	(32%)[32%](3%)[3%]	TE-Special Code	1.070	1.750	2.365	3.090	3.660	7.425	10.620	14.275	17.840
40%	60%	TE-Revised Simplex	2.085	4.268	6.293	8.746	11.254	24.265	39.140	51.810	69.668
(36%)[4%]	(28%)[28%](2%)[2%]	TE-Special Code	1.245	2.085	3.030	3.863	4.798	9.420	14.710	19.215	25.676
50%	50%	TE-Revised Simplex	2.287	4.303	6.797	9.185	11.315	24.875	42.257	55.553	72.420
(45%)[5%]	(24%)[24%](1%)[1%]	TE-Special Code	1.530	2.455	3.720	4.720	5.733	11.995	19.053	24.771	31.998

Note: ( ) [ ] with Efficient DMUs indicate the percentages of E and E', while ( ) [ ] ( ) [ ] with Inefficient DMUs indicate the percentages of IE, IE', IF and IF', respectively.

Table 2: Ratio of CPU time (%) [TE-Special Code/TE-Revised Simplex]

Efficient DMUs (%)		Inefficient DMUs (%)		Numbers of DMUs										
E	U E'	IE	U IE'	U IF'	U IF'	20	40	60	80	100	200	300	400	500
10%		90%				43.73	33.18	25.81	24.02	21.71	16.79	15.16	14.59	14.20
20%		80%				46.53	37.99	32.69	29.71	28.44	23.31	22.99	21.32	20.57
30%		70%				53.10	41.87	39.42	37.49	33.90	30.99	29.65	29.14	28.22
40%		60%				59.71	48.85	48.15	44.17	42.63	38.82	37.58	37.09	36.85
50%		50%				66.90	57.05	54.97	51.39	50.67	48.22	45.09	44.59	44.18

Computational results of this simulation study are summarized in Tables 1 and 2. The numbers in Table 1 indicate average CPU seconds, each of which is measured by dividing the total CPU seconds by the total number of DMUs. For instance, the time shown in the last column (i.e., the one with No. of DMUs= 500) is the average CPU time to solve 5000 (10 × 500) DMUs where 10 is the number of replications and 500 is the number of DMUs. Table 2 presents the ratio of CPU time, expressed in a percentage form, that compares the average CPU time of the TE-special code with that of the dumb code (the TE-revised simplex code). As described in Table 2, at least 35% savings are yielded by the TE-special code. Moreover, Table 2 depicts that there is a decreasing trend in the ratio of CPU time as the number of DMUs and/or the percentage of inefficient DMUs increase. The finding indicates clearly that the performance of the TE-special code becomes very efficient particularly when a data set becomes very large.

#### 4.2 Simulation study for AE measurement

Using the same data set in Section 4.1, this study compared the performance of two DEA codes related to AE measurement. The first code (named “AE-revised simplex”) used the ordinary revised simplex method to solve (4). The DEA coded was a dumb code that always started from scratch to measure each DMU, following the observed order of DMUs. The second code (named “AE-special code”) used the algorithmic strategy concerning AE measurement depicted in Figure 1.

DEA/AR used in this simulation study incorporates the following range of AR to restrict its dual variables:

- d. Range of AR:  $0 \leq u_2/u_1 \leq 1, 0 \leq v_2/v_1 < 1$  and  $1 \leq v_3/v_1 \leq 100$ .

Computational results of the simulation study are summarized in Table 3 and Table 4. The two tables correspond to Table 1 and Table 2, respectively. Table 3 depicts that at least 35% savings are yielded by the AE-special code. This simulation study also finds that there is a decreasing trend in the ratio of CPU time as the number of DMUs and/or the percentage of inefficient DMUs increase. Thus, the finding also indicates that the performance of the special algorithm for measuring AE is increasingly improved as the size of a data set becomes large.

Table 3: Comparison between two DEA codes for measuring AE (average CPU seconds)

Efficient DMUs (%)		Inefficient DMUs (%)		Numbers of DMUs									
E	U E'	IE	U IE' U IF U IF'	20	40	60	80	100	200	300	400	500	
10%		90%		AE-Revised Simplex	1.733	2.935	4.984	6.923	9.430	21.303	33.712	50.800	71.271
(9%)	[1%]	(40%)	[40%](5%)[5%]	AE-Special Code	0.744 (2)	0.956 (3)	1.262 (5)	1.632 (6)	2.010 (8)	3.390 (16)	4.686 (25)	6.820 (32)	9.341 (41)
20%		80%		AE-Revised Simplex	1.834	3.836	6.078	8.247	10.835	27.036	38.564	60.428	78.122
(18%)	[2%]	(36%)	[36%](4%)[4%]	AE-Special Code	0.837 (4)	1.430 (7)	1.950 (11)	2.405 (14)	3.025 (16)	5.973 (33)	8.127 (48)	11.853 (65)	14.834 (84)
30%		70%		AE-Revised Simplex	2.136	4.473	6.480	8.984	11.875	27.554	42.980	61.218	82.188
(27%)	[3%]	(32%)	[32%](3%)[3%]	AE-Special Code	1.113 (6)	1.838 (11)	2.507 (17)	3.306 (22)	3.953 (26)	8.093 (49)	11.682 (73)	16.416 (98)	21.408 (125)
40%		60%		AE-Revised Simplex	2.210	4.567	6.796	9.533	12.379	27.904	46.968	64.672	90.568
(36%)	[4%]	(28%)	[28%](2%)[2%]	AE-Special Code	1.295 (7)	2.189 (15)	3.212 (22)	4.133 (27)	5.182 (33)	10.267 (70)	16.181 (105)	22.097 (140)	30.811 (175)
50%		50%		AE-Revised Simplex	2.424	4.604	7.341	10.011	12.446	28.606	50.708	60.441	94.146
(45%)	[5%]	(24%)	[24%](1%)[1%]	AE-Special Code	1.591 (9)	2.578 (19)	3.943 (27)	5.050 (35)	6.191 (43)	13.074 (83)	20.958 (122)	28.486 (165)	38.397 (210)

Note: ( ) [ ] with Efficient DMUs indicate the percentages of E and E', while ( ) [ ] ( ) [ ] with Inefficient DMUs indicate the percentages of IE, IE', IF and IF', respectively. The (number) under AE-Special Code indicates the number of DMUs in EP.

Table 4: Ratio of CPU time (%) [AE-Special Code/AE-Revised Simplex]

Efficient DMUs (%)		Inefficient DMUs (%)				Numbers of DMUs								
E	U E'	IE	U IE'	U IF	U IF'	20	40	60	80	100	200	300	400	500
10%		90%				42.90	32.55	25.32	23.57	21.31	15.71	13.87	13.42	13.10
20%		80%				45.65	37.28	32.08	29.16	27.91	22.09	21.07	19.61	18.98
30%		70%				52.09	41.08	38.68	36.80	33.28	29.37	27.17	26.81	26.04
40%		60%				58.58	47.93	47.25	43.35	41.85	36.79	34.45	34.12	34.01
50%		50%				65.63	55.98	53.71	50.44	49.74	45.70	41.33	41.02	40.78

## 5. Conclusion and Future Extensions

The objective of this article is to describe an efficiently designed framework to measure TE and AE, using the two DEA models. An important feature of this algorithmic strategy is that it separates all the DMUs into six distinct subsets. Each subset is solved by a different algorithm. As a result of the algorithmic framework, the proposed approach can measure TE and AE with less CPU time and computational efforts than the ordinary revised simplex method. Experimental simulation studies have confirmed the superiority of the specially designed algorithm.

As a future extension, this study proposes the use of the parallel processing incorporated in a supercomputer system for DEA. That is, the DEA data set can be classified into several subsets, each of which can be solved by a parallel processing procedure. Thus, an efficient algorithm including both the special DEA algorithm and the architecture of the supercomputer can solve extremely large DEA problems. Furthermore, the research demonstrates that DEA is a perfect application for such parallel processing system. This is an important future task.

A shortcoming of this research is that our experimental simulation studies examine the performance of the proposed DEA algorithm, using only data sets of two outputs and three inputs. It is important to develop a simulation framework by which more than three outputs and/or inputs are generated. However, this task is very difficult from the perspective of controlling the types of DMUs. This issue is another important assignment, as well.

Finally, it is hoped that this study makes a small contribution regarding DEA algorithmic development and its applications.

## References

- [1] Charnes, A., W.W. Cooper and E. Rhodes: Measuring the Efficiency of Decision Making Units, *European Journal of Operational Research*, Vol. 2 (1978), 429–444.
- [2] Charnes, A., W.W. Cooper and R.M. Thrall: Classifying and Characterizing Efficiencies and Inefficiencies in Data Envelopment Analysis, *Operations Research Letters*, Vol. 5 (1986), 105–110.
- [3] Charnes, A., W.W. Cooper and R.M. Thrall: A Structure of Characterizing and Classifying Efficiencies and Inefficiencies in Data Envelopment Analysis, *Journal of Productivity Analysis*, Vol. 2 (1991), 197–237.
- [4] Charnes, A., W.W. Cooper, Q.L. Wei and Z.M. Huang: Cone Ratio Data Envelopment

- Analysis and Multi-Objective Programming, *International Journal of Systems Science*, Vol. 20 (1989), 1099–1118.
- [5] Chang, Y.L. and T. Sueyoshi: An Interactive Application of Data Envelopment Analysis in Microcomputers, *Computer Science in Economics and Management*, Vol. 4 (1991) 51–64.
- [6] Farrel, M.J.: The Measurement of Productive Efficiency, *Journal of Royal Statistics Society*, Vol. 120 (1957), Series A, General, 253–281.
- [7] Koopmans, T.C.: *Activity Analysis of Production and Allocation*, Cowles Commission Monograph 13, John Wiley, New York, (1951) 33–97.
- [8] Pareto, V.: *Manuale di economia politica, con una introduzione alla scienza sociale*, Scocieta Editrice Libreria, Milan, Italy (1906).
- [9] Seiford, L.H.: A Bibliography of Data Envelopment Analysis (1978–1990), *DEA Bibliography*, Working Paper, Department of Industrial Engineering and Operations Research, The University of Massachusetts, Amherst, MA.
- [10] Sueyoshi, T.: A Special Algorithm for an Additive Model in Data Envelopment Analysis, *Journal of Operational Research Society*, Vol. 41 (1990), 249–257.
- [11] Sueyoshi, T.: A Study on Efficiency Analysis Using DEA (in Japanese), *Communication of the Operations Research Society of Japan*, Vol. 33 (1990), 167–173.
- [12] Sueyoshi, T.: Measuring Technical, Allocative and Overall Efficiency Using DEA Algorithm, *Journal of the Operational Research Society*, Vol. 42 (1992), 141–155.
- [13] Sueyoshi, T., and Y.L. Chang: Efficient Algorithm for Additive and Multiplicative Models in Data Envelopment Analysis, *Operations Research Letters*, Vol. 8 (1989), 205–213.
- [14] Thompson, R.G., F.D. Singleton, R.M. Thrall and B.A. Smith: Comparative Site Evaluation for Locating a High-Energy Physics Lab in Texas, *Interface*, Vol. 16 (1986), 35–49.
- [15] Thrall, R.M.: Overview and Recent Developments in DEA: The Mathematical Programming Approach, Working Paper No. 66, Jesse H. Jones Graduate School of Admission, Rice University (February, 1989).
- [16] Tone, K.: Efficiency Analysis for Decision Making Units (in Japanese), *Communication of the Operations Research Society of Japan*, Vol. 32 (1987), 800–803, and Vol. 33 (1988), 45–48 and 95–99.

Toshiyuki Sueyoshi  
The Ohio State University  
College of Business  
School of Public Policy & Management  
1775 College Road  
Columbus, Ohio 43210  
U.S.A.