Pekka J. Korhonen – Pyry-Antti Siitari

USING LEXICOGRAPHIC PARAMETRIC PROGRAMMING FOR IDENTIFYING EFFICIENT UNITS IN DEA

HELSINKI SCHOOL OF ECONOMICS
WORKING PAPERS
W-381

Pekka J. Korhonen – Pyry-Antti Siitari

USING LEXICOGRAPHIC PARAMETRIC PROGRAMMING FOR IDENTIFYING EFFICIENT UNITS IN DEA

Quantitative Methods in Economics and
Management Science

December
2004

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
WORKING PAPERS
W-381

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
PL 1210
FIN-00101 HELSINKI
FINLAND

Pekka Korhonen
Helsinki School of Economics
Runeberginkatu 14-16, 00100 Helsinki, Finland
Email: pekka.korhonen@hkkk.fi

The research was supported by the Academy of Finland.

Acknowledgements: We would like to thank Prof. Jose Dulá for useful discussions and providing us with the data files we used in our performance tests.

ISSN 1235-5674
ISBN 951-791-902-6 (Electronic working paper)

Helsinki School of Economics -
HeSE print 2005

ABSTRACT

In this paper, we propose the use of lexicographic parametric programming to recognize efficient units in Data Envelopment Analysis (DEA). By using the parameterization of the rhs - vector of the envelopment problem, we obtain the efficiency curve which is traversing through the efficient frontier from unit to unit. The units in the basis with any parameter value are efficient and the unit dominated by a point on an efficient facet is inefficient. The second objective is needed to check that efficient curve on the boundary of the efficient frontier stays on the boundary.

Keywords: Efficiency Analysis, Data Envelopment Analysis, Lexicographic, Parametric Programming, Computational Aspects

1 Introduction

Charnes, Cooper and Rhodes [1978] developed Data Envelopment Analysis (DEA) for evaluating the relative efficiency of comparable units called Decision Making Units (DMUs) essentially performing the same task using similar multiple inputs to produce similar multiple outputs. The units are assumed to operate under similar conditions. Based on information about existing data on the performance of the units and some preliminary assumptions, DEA forms an empirical efficient surface (frontier). If a DMU lies on the surface, it is a referred to as an efficient unit, otherwise inefficient. DEA also provides efficiency scores and reference set for inefficient DMUs. The efficiency scores are used in practical applications as performance indicators of the DMUs. The reference set for inefficient units consists of efficient units and determines a virtual unit on the efficient surface. The virtual unit can be regarded as a target unit for the inefficient unit.

The target unit is found in DEA by projecting an inefficient DMU radially[1] to the efficient surface. To check the efficiency of a unit, and to find the reference set and the efficiency score for inefficient units requires the solving of an LP-model. The "standard" basic algorithm solves iteratively an LP-model for each unit separately. At each iteration, the rhs - vector and one column (direction vector) in the coefficient matrix has to be updated. The optimal basis of the previous iteration is not valid for the next iteration as such. Actually, it is not necessary to solve an LP – model for each unit, because all units in an optimal basis for some unit are efficient, and those units are not needed further investigation. The approach is usable in small problems[2], but is computationally ineffective in large scale problems.

Additional problems are caused by the weakly efficient solutions. The original problem formulation (Charnes *et al.* [1978]) sometimes led to weakly efficient solutions, but the authors recognized the problem and reformulated the model in Charnes *et al.* [1979] by using a so-called non-Archimedean infinitesimal in the model. There are two ways to deal with the infinitesimal in the model: 1) to replace it by a small number or 2) to use a lexicographic approach. When a small number is

[1] Term "radial" means that an efficient frontier is tried to reach by increasing the values of the current outputs or decreasing the values of the current inputs, or doing the both ones simultaneously in the same proportion.

[2] Dulá and López [2002] called small problems the ones consisted of less than 500 units.

used, it is important that it is properly chosen. If it is too big, some efficient units are diagnosed inefficient. If it is too small, weakly efficient units are recognized efficient. The lexicographic approach is a better way. When the optimal solution of an LP-model is not unique, the second objective is used to check whether the unit is efficient or not (see, e.g. Steuer [1986], p. 445).

When the number of the units is large, let us say many ten thousands or even hundreds of thousands, computational aspects are important. Such problems appear when, for example, all high-schools or hospitals in Europe are evaluated, or when the efficiency analysis is made at an individual level. The straightforward approach to formulate an LP-model for each unit with an unknown status does not work. It is too time-consuming. Fortunately, the structure of the DEA-model makes it possible to develop special techniques for large-scale problems.

There are only few authors who have studied computational problems in DEA. In the paper by Ali [1993], the main idea was to restrict the basis entry. The basis always consists of a set of existing (efficient) units. When a unit is diagnosed inefficient, the corresponding column can be dropped from the set of potential basic vectors. In most cases, the technique clearly reduced computation time. Dulá and Helgason [1996] proposed the solving of the problem in two phases. In phase I, the extreme point solutions of the polytope consisting of all units in the data set are defined. The efficiency scores of the other vectors are computed in phase II by using the minimal set of potential basic vectors, i.e. efficient extreme units. The idea was further developed in the paper by Dulá *et al.* [1997]. The most recent developments by Dulá and his associates are presented in Dulá and López [2002]. Because the computing time as the function of the units increases more than linearly, Barr and Durchholz [1997] proposed the partition of the problem. The idea makes it possible to first identify the set of the efficient units in a small data set, and then to use those units to build a set of potential basic vectors. The union of those sets consists of all efficient units, but usually also inefficient units.

In this paper, we propose the use of lexicographic parametric programming (Korhonen and Halme [1996]) to classify the units efficient and inefficient. Using that technique, we may move from unit to unit along an efficiency curve. The units entering the basis are recognized efficient and all units dominated by an efficient facet are inefficient. The move on the curve is terminated, when the end unit is diagnosed. The lexicographic parametric programming is needed to guarantee that the curve will stay on the efficient frontier also in case when it reaches the boundary.

After recognizing all efficient units, we may compute the scores of the inefficient units in the second phase like proposed by Barr and Durchholz [1997] and Dulá *et al.* [1997]. The number of the columns in these LP-problems is usually much smaller than in the original problem.

The paper is given in six sections. In the next section, we review the main ideas of lexicographic parametric linear programming. In Section 3, the basic DEA models and necessary theory are represented and Section 4 consists of the main principles of the procedure and illustrated with a numerical example. Computational results are given and discussed in Section 5. Section 6 concludes the paper with some remarks.

2 Lexicographic Parametric Programming

Consider the following problem ($p > 1$):

$$\begin{aligned} & lex\ max\ \{\boldsymbol{c}^1\boldsymbol{x}, ..., \boldsymbol{c}^p\boldsymbol{x}\} \\ & \text{s.t.} \\ & \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \\ & \quad \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \tag{2.1}$$

where $\boldsymbol{c}^k$, $k = 1, 2, \ldots, p$, are $(1 \times n)$ -vectors of the coefficients of the objective functions , $\mathbf{A}$ is an $(m \times n)$ – matrix of coefficients, and the rhs-vector $\boldsymbol{b}$ is an $(m \times 1)$ – vector.

For simplicity, we assume rank($\mathbf{A}$) = m. Notation "*lex*" refers to lexicographic optimization. The solution of the lexicographic optimization problem has to be optimal for each of the following models.

The model for the objective function $\boldsymbol{c}^1\boldsymbol{x}$:

$$\begin{aligned} & max\ \mathbf{c}^1\boldsymbol{x} \\ & \text{s.t.} \\ & \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \\ & \quad \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned} \tag{2.2a}$$

The model for the objective functions $\boldsymbol{c}^k\mathbf{x}$, $k = 2, 3, \ldots, p$:

$$\begin{aligned} & max\ \boldsymbol{c}^k\mathbf{x} \\ & \text{s.t.} \\ & \quad \boldsymbol{c}^i\boldsymbol{x} = \boldsymbol{q}_i^*,\ i = 1, 2, \ldots, k\text{-}1 \\ & \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \\ & \quad \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned} \tag{2.2b}$$

where $\boldsymbol{q}_i^*$ is the optimal value of the lexicographic optimization problem of the objective function $\boldsymbol{c}^i\mathbf{x}$, $i = 1,2, \ldots, k$-1.

Consider the optimal solution $\boldsymbol{x}^*$ of the lexicographic optimization problem. Let $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] = [\mathbf{B}, \mathbf{N}]$, where $\mathbf{B}$ is an optimal basis corresponding to the optimal solution $\boldsymbol{x}^*$, and $\mathbf{N}$ consists of the non-basic columns of $\mathbf{A}$. We denote $\boldsymbol{z}^k_j = \boldsymbol{c}^k_B\,\mathbf{B}^{-1}\boldsymbol{a}_j$.

Definition 1. The solution $\boldsymbol{x}^*$ is the optimal solution of lexicographic optimization problem (2.1) if it is feasible and the following optimal conditions are fulfilled:

$$z^1_j - c^1_j \geq 0,\ j \in \mathrm{R} \text{ and} \tag{2.3}$$

$z^k_j - c^k_j \geq 0$, for all $k \in \{2, \ldots, p\}$ and $j \in \mathrm{R}$ for which $z^i_j - c^i_j = 0$ for $i = 1, \ldots, k$-1, where R is the index set of all non-basic variables.

For instance, Steuer [1986, pp. 292-296] has further discussed the use of the optimality conditions with respect to lexicographic goal programming. He also described a lexicographic simplex method, which leads to the optimal solution of the lexicographic model. See, also Sawaragi *et al.* [1985, p. 276].

Korhonen and Halme [1996] formulated a lexicographic parametric programming approach for searching non-dominated solutions in Multiple Objective Linear Programming (MOLP) problems:

$$\begin{aligned} &\mathit{lex\ max}\ \{\boldsymbol{c}^1\boldsymbol{x}, ..., \boldsymbol{c}^p\boldsymbol{x}\} \\ &\text{s.t.} \\ &\qquad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} + t\Delta\boldsymbol{b} \\ &\qquad \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \tag{2.4}$$

where $\Delta\boldsymbol{b}$ is a reference direction vector and t a parameter.

In lexicographic parametric programming, the leaving variable x_r is chosen as usually in parametrizing the rhs-vector in LP. The entering variable x_s is chosen using the following procedure which guarantees that the basis remains optimal (see, Korhonen and Halme [1996]):

Choose $s \in \mathrm{L_u}$, $\mathrm{u} = \max\limits_{k \leq p} \{k \mid \mathrm{L}_k \neq \varnothing\}$, where the index sets L_k, $k = 0, 2, \ldots, p$, are defined as

$$\mathrm{L}_0 = \{j \mid x_j \text{ is non-basic and } y_{rj} < 0\} \tag{2.5}$$

$$\mathrm{L}_k = \begin{cases} \varnothing, \text{ if } |\mathrm{L}_{k\text{-}1}| = 1 \text{ and } k \geq 1 \\ \{h \mid (z_h^k - c_h^k)/\, y_{rh} = \max\limits_{j \in L_{k\text{-}1}} [\, (z_j^k - c_j^k)/y_{rj}]\}, \text{ if } |\mathrm{L}_{k\text{-}1}| > 1 \text{ and } k \geq 1 \end{cases}$$

where $\boldsymbol{y}_j = [y_{1j}, y_{2j}, \ldots, y_{nj}]^{\mathrm{T}} = \mathbf{B}^{-1}\boldsymbol{a}_j$.

In lexicographic parametric programming, we proceed as usually in parametric linear programming until the choice of the entering variable is not uniquely defined. In this case, the second objective is chosen so that the solution remains lexicographically optimal (see, for more details in Korhonen and Halme [1996]).

3 Theoretical Considerations

3.1 Basic Data Envelopment Models

Assume we have n DMUs each consuming m inputs and producing p outputs. Let $\mathbf{X}$ be an $(m \times n)$ - matrix and $\mathbf{Y}$ be a $(p \times n)$ - matrix consisting of non-negative elements, containing observed input and output measures for the DMUs, respectively. We denote by $\boldsymbol{x}_j$ (the *j*th column of **X**) the vector of inputs consumed by DMU$_j$, and by x_{ij} the quantity of input *i* consumed by DMU*j*. A similar notation is used for outputs. We further assume that $\boldsymbol{x}_j \neq \boldsymbol{0}$ and $\boldsymbol{y}_j \neq \boldsymbol{0}$, $j = 1, 2, \ldots, n$, and that every unit in the data set is unique; there are no duplicates. Furthermore, we denote $\boldsymbol{1} = [1, ..., 1]^T$.

The traditional CCR-models, as introduced by Charnes *et al.* [1978] are fractional linear programs which can easily be formulated and solved as linear programs. Those models are so-called constant returns to scale models. Later Banker, Charnes and Cooper [1984] developed the so-called BCC models with variable returns to scale. The CCR and BCC models are the basic model types in DEA. Those basic models can be presented in a primal or dual form. Which one is primal or dual varies. It is better to call them multiplier and envelopment models accordingly. The multiplier model provides information on the weights of inputs and outputs which are interpreted as prices in many applications. Instead, the envelopment models provide the user with information on the lacks of outputs and the surplus of inputs of a unit. Moreover, the envelopment model characterizes the reference set for inefficient units.

Without losing generality, we will consider a DEA-model by using a general directional vector $\boldsymbol{w} = \begin{pmatrix} \boldsymbol{w}^y \\ \boldsymbol{w}^x \end{pmatrix} \geq \boldsymbol{0}$, $\boldsymbol{w} \neq \boldsymbol{0}$ (discussion on directional distance functions, see Chambers *et al.* [1998]). Halme *et al.* [1999] called the model a general combined model. Input- and output-oriented models are the special cases of that model.

Consider the following general DEA - formulation in the so-called envelopment form:

$$
\begin{aligned}
&\max \;\; Z = \sigma + \varepsilon(\boldsymbol{1}^T\boldsymbol{s}^+ + \boldsymbol{1}^T\boldsymbol{s}^-) \\
&\text{s.t.} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.1)\\
&\quad \mathbf{Y}\boldsymbol{\lambda} - \sigma\boldsymbol{w}^y - \boldsymbol{s}^+ = \boldsymbol{y}_0 \\
&\quad \mathbf{X}\boldsymbol{\lambda} + \sigma\boldsymbol{w}^x + \boldsymbol{s}^- = \boldsymbol{x}_0 \\
&\qquad \boldsymbol{\lambda} \in \Lambda \\
&\qquad \boldsymbol{\lambda}, \boldsymbol{s}^-, \boldsymbol{s}^+ \geq \boldsymbol{0} \\
&\qquad\qquad \varepsilon > 0 \qquad (\text{"Non-Archimedean"}),
\end{aligned}
$$

where $\boldsymbol{x}_0$ is the input-vector and $\boldsymbol{y}_0$ is the output-vector of a DMU under consideration and

$$
\Lambda = \begin{cases}
\{\boldsymbol{\lambda} \mid \boldsymbol{1}'\boldsymbol{\lambda} = 1, \boldsymbol{\lambda} \geq \boldsymbol{0}\} & \text{for variable returns to scale model (Banker et al. [1984])} \\
\{\boldsymbol{\lambda} \mid \boldsymbol{1}'\boldsymbol{\lambda} \leq 1, \boldsymbol{\lambda} \geq \boldsymbol{0}\} & \text{for non-increasing returns to scale model} \\
\{\boldsymbol{\lambda} \mid \boldsymbol{1}'\boldsymbol{\lambda} \geq 1, \boldsymbol{\lambda} \geq \boldsymbol{0}\} & \text{for non-decreasing returns to scale model} \\
\{\boldsymbol{\lambda} \mid \boldsymbol{\lambda} \geq \boldsymbol{0}\} & \text{for constant returns to scale model (Charnes } et\ al.\text{ [1978])}
\end{cases}
$$

The epsilon constraint ($\varepsilon > 0$) was not in the original formulation by Charnes *et al.* [1978]. One year later the authors published the revised model (Charnes *et al.* [1978]), in which the importance of the ε - constraint was recognized. Without that constraint, the solution of model (3.2) may be a weakly-efficient.

In the combined model, $\boldsymbol{w}^y = \boldsymbol{y}_0$ and $\boldsymbol{w}^x = \boldsymbol{x}_0$. In the input-oriented $\boldsymbol{w}^y = \boldsymbol{0}$ and $\boldsymbol{w}^x = \boldsymbol{x}_0$ model, and $\boldsymbol{w}^x = \boldsymbol{0}$ and $\boldsymbol{w}^y = \boldsymbol{y}_0$ in the output-oriented model. A DMU is efficient if and only if (iff) the optimal value Z* of model (2.1) equals 0. All slack variables $\boldsymbol{s}^-$, $\boldsymbol{s}^+$ equal zero, too. Otherwise, the DMU is inefficient (Charnes *et al.* 1994). The value of σ - called an inefficiency score - at the optimum is denoted by σ^*. When the unit is efficient, $\sigma^* = 0$; otherwise $\sigma^* > 0$. Note that Z* is not necessarily equal to σ^*. For weakly-efficient solutions, Z* > 0, but $\sigma^* = 0$.

In this paper there is no need to emphasize the different roles of inputs and outputs. Therefore, we simply denote $\boldsymbol{u} = \begin{pmatrix} \boldsymbol{y} \\ -\boldsymbol{x} \end{pmatrix}$ and $\mathbf{U} = \begin{pmatrix} \mathbf{Y} \\ -\mathbf{X} \end{pmatrix}$. We call $\boldsymbol{u}$ an input/output-vector, although to be precise an input/output-vector is $\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{x} \end{pmatrix}$. In order to avoid specifying a value of $\varepsilon > 0$, we use the lexicographic formulation for model (3.1) (see, e.g. Cooper *et al.* [2000, p. 44]:

$$\begin{aligned} &\textit{lex max} \ \{\sigma, \boldsymbol{1}^T\boldsymbol{s}\} \\ &\text{s.t.} \\ &\quad \mathbf{U}\boldsymbol{\lambda} - \sigma\boldsymbol{w} - \boldsymbol{s} = \boldsymbol{u}_0 \\ &\quad \boldsymbol{\lambda} \in \Lambda \\ &\quad \boldsymbol{\lambda}, \boldsymbol{s} \geq \boldsymbol{0} \end{aligned} \tag{3.2}$$

where $\boldsymbol{u}_0$ is the input/output-vector of a DMU and $\boldsymbol{s} = \begin{pmatrix} \boldsymbol{s}^+ \\ \boldsymbol{s}^- \end{pmatrix}$.

Notation "*lex max*" means that we first solve (3.2) using σ as an objective function. In case, the optimal solution σ^* is not unique, we formulate a new model by adding the constraint $\sigma = \sigma^*$ into the model (3.2) and solve it by using $\boldsymbol{1}^T\boldsymbol{s}$ as the objective function.

We denote $r = m + p$, and define the set $\mathrm{K} = \{\boldsymbol{u} \mid \boldsymbol{u} = \mathbf{U}\boldsymbol{\lambda}, \boldsymbol{\lambda} \in \Lambda\}$.

The definition of efficiency and weak efficiency together with the related definition for extreme efficiency can be given in the following form:

Definition 2. A point $\boldsymbol{u}^* \in \mathrm{K}$ is *efficient* iff (if and only if) there does not exist another $\boldsymbol{u} \in \mathrm{K}$ such that $\boldsymbol{u} \geq \boldsymbol{u}^*$, and $\boldsymbol{u} \neq \boldsymbol{u}^*$.

Definition 3. A point $\boldsymbol{u}^* \in \mathrm{K}$ is *weakly efficient* iff there does not exist another $\boldsymbol{u} \in \mathrm{K}$ such that $\boldsymbol{u} > \boldsymbol{u}^*$.

Definition 4. A point $\boldsymbol{u}^* \in \mathrm{K}$ is *efficient extreme* iff there does not exist other points $\boldsymbol{u}_1 \in \mathrm{K}$, $\boldsymbol{u}^* \neq \boldsymbol{u}_1$, and $\boldsymbol{u}_2 \in \mathrm{K}$, $\boldsymbol{u}^* \neq \boldsymbol{u}_2$, such that $\boldsymbol{u}^* \leq \lambda\boldsymbol{u}_1 + (1-\lambda)\boldsymbol{u}_2$ for some $\lambda \in [0, 1]$.

The set of efficient points is denoted by K^{E}, the set of weakly efficient points by K^{W} and the set of efficient extreme points by K^{EE}.

3.2 Checking Efficiency of Next Point

For simplicity, but without loosing generality, we assume $\boldsymbol{u}_0 \in \mathrm{K}^{\mathrm{E}}$. We may use the same procedure, but for conceptual reasons we start from an efficient point. Consider the efficiency of a unit DMU$_j$, $j \in \{1, 2, \ldots, n\}$, We formulate the following lexicographic parametric problem:

$$
\begin{aligned}
& \mathit{lex\ max} \quad \{\sigma, \boldsymbol{1}^T\boldsymbol{s}\} \\
& \text{s.t.} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.3) \\
& \qquad \mathbf{U}\boldsymbol{\lambda} - \sigma\boldsymbol{w} - \boldsymbol{s} = \boldsymbol{u}_0 + t(\boldsymbol{u}_j - \boldsymbol{u}_0) \\
& \qquad\qquad \boldsymbol{\lambda} \in \Lambda \\
& \qquad\qquad \boldsymbol{\lambda},\ \boldsymbol{s} \geq \boldsymbol{0} \\
& \qquad\qquad t\text{: } 0 \to b,
\end{aligned}
$$

where $\boldsymbol{u}_j$ is an input/output-vector corresponding to DMU$_j$ and $b \leq 1$ is the value of t at which the status of $\boldsymbol{u}_j$ is diagnosed.

The search may be stopped at the moment when $\boldsymbol{u}_1$ is entering the optimal basis with ($t \leq 1$). DMU$_1$ can be recognized efficient. The search terminates in sure, when $t = 1$. If DMU$_1$ may be selected to the optimal basis at latest when $t = 1$, it is a sufficient condition that DMU$_1$ is efficient. [3]

In lexicographic parametric programming, we proceed as usually in parametric programming until the choice of the entering variable is not uniquely defined. In this case, we use the second objective to keep the basis lexicographically optimal. The method is developed in Korhonen and Halme [1996].

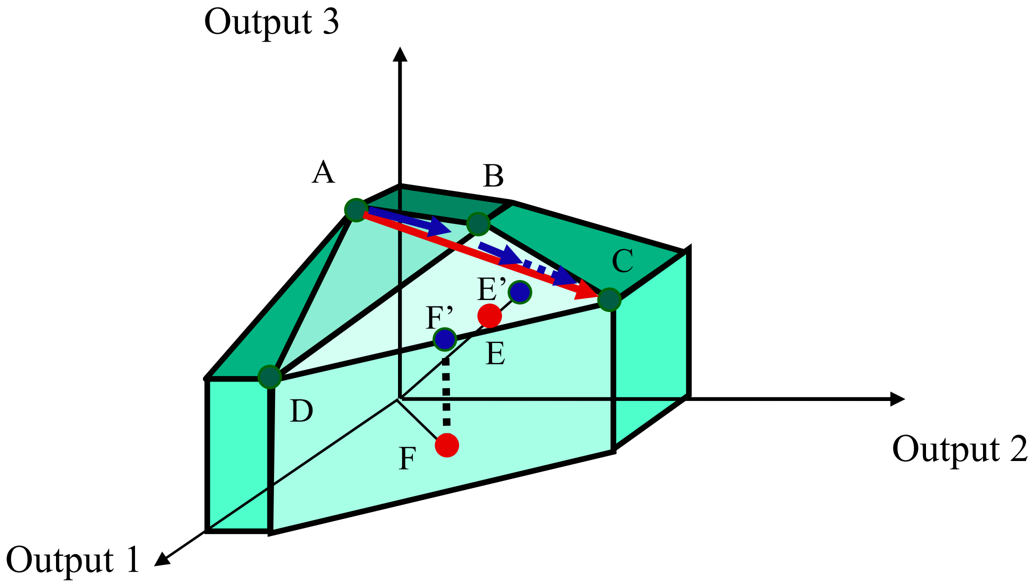

Figure 1: Illustration of Parameterization

We illustrate how the efficiency of units A, B, C, D, E, and F in Figure 1 will be checked. We assume three outputs and one identical input. Numerical computations are shown in Section 4.2 (In this example, for simplicity we use these capital letters to refer to units.)

[3)] Unfortunately, it is not a necessary condition. If the unit lies on an efficient facet, but is not an extreme point solution, it is efficient, but not in any optimal basis. This special case is easy to check. If the current basis is lexicographically optimal, $\sigma^* = 0$ and $\boldsymbol{1}^T\boldsymbol{s}^* = 0$, the unit is efficient (but non-extreme).

We start the search from A. Because the optimal basis consists of A, we know that it is efficient. Consider next unit C. From point A we start to move on line AC. The projection of this vector is called an efficiency curve and initially it goes along facet ABD. Units B and D enter the optimal basis with some $t \in [0, 1]$ (actually $t = 0$), and thus they are recognized efficient. When we are crossing an edge BD, unit A leaves the basis and unit C enters. We may stop the search on the edge BD, because C can now diagnosed efficient.

There is no need to perform a special search for unit E, because its projection E' lies on facet BCD, indicating that E is inefficient. No special computation is needed. Information is available in the simplex tableau (see, for a numerical example in Section 4.2).

To investigate the efficiency of unit F, we start the search from the point on the edge at which we stopped the search for C. When this direction is projected onto the efficient frontier it goes along the facet BCD, until it reaches the edge DC. At this point, we have to use a lexicographic rule to stay on the efficient frontier, i.e. on the edge DC. Unit B leaves the basis, and Output 3 enters. From this new optimal tableau, we may see that F is inefficient (weakly efficient) in the same way as E from the previous basis.

4 Development of the Procedure

4.1 Description of the Procedure

In this section, we describe the procedure and illustrate it with a numerical example.

Step 0: Initialization

Choose the projection vector $\boldsymbol{w} \geq \boldsymbol{0}$ and $\boldsymbol{w} \neq \boldsymbol{0}$. We recommend that the vector $\boldsymbol{w} > \boldsymbol{0}$ is used, because a need to apply a lexicographic rule is not so likely. To avoid computational difficulties, it is important that each element of $\boldsymbol{w}$ is of the same magnitude as the corresponding inputs/outputs. A simple rule to choose $w_i = \sum_{j=1}^{n} \frac{|u_{ij}|}{n}$.

Choose an initial unit DMU$_k$, $k \in \{1, 2, \ldots, n\}$, set $\boldsymbol{u}_0 := \boldsymbol{u}_k$ and project $\boldsymbol{u}_0$ onto the efficient frontier by using the formula (3.2):

Define the following index sets:

- K^{EE} is the index set of the units diagnosed efficient extreme. Set K^{EE} := {the indices of the units in the current optimal basis}
- K^{EN} is the index set of the units diagnosed efficient but non-extreme ($K^E = K^{EE} \cup K^{EN}$). Set $K^{EN} := \varnothing$.
- K^W is the index set of the units diagnosed weakly efficient. Set $K^W := \varnothing$.
- K^I is the index set of the units diagnosed inefficient. Set $K^I := \varnothing$.

- K^U is the ordered index set of the units with an unknown status. Set $K^U := \{1, 2, \ldots, n\} \cap - \{K^{EE} \cup \{k\}\}$

If $k \in K^{EE}$ then go to **Step 1**.

Else diagnose the unit k:

If the value σ in the optimal basis is positive ($\sigma > 0$), the unit k is inefficient.

Define $K^I := \{k\}$.

If the value σ is zero but any of the slack-variables $\boldsymbol{s}$ has a positive value ($\sigma = 0$ and $\boldsymbol{1}^T\boldsymbol{s} \neq 0$), the unit k is weakly efficient.

Define $K^W := \{k\}$.

If the value σ is zero and all of the slack-variables $\boldsymbol{s}$ are also zero ($\sigma = 0$ and $\boldsymbol{1}^T\boldsymbol{s} = 0$), the unit k is efficient but non-extreme.

Define $K^{EN} := \{k\}$.

Drop the column corresponding to unit k from the matrix $\mathbf{U}$. Go to **Step 1**.

Step 1: Model Formulation for the Checking Efficiency of a Unit

If $K^U = \varnothing$ then **Stop**.

Choose a new unit DMU_k, $k \in K^U$, for consideration[4]. Set $t_0 := 0$.

Consider the parametric programming formulation:

$$\begin{aligned} &\textit{lex max} \quad \{\sigma, \boldsymbol{1}^T\boldsymbol{s}\} \\ &\text{s.t.} \\ &\quad \mathbf{U}\boldsymbol{\lambda} - \sigma\boldsymbol{w} - \boldsymbol{s} = \boldsymbol{u}_0 + t\boldsymbol{\delta} \\ &\quad\quad \boldsymbol{\lambda} \in \Lambda \\ &\quad\quad \boldsymbol{\lambda}, \mathbf{s} \geq \boldsymbol{0} \\ &\quad\quad t: 0 \rightarrow 1, \end{aligned} \qquad (3.4)$$

where $\boldsymbol{\delta} = \boldsymbol{u}_k - \boldsymbol{u}_0$. (Actually, the updating of the parameter vector in the simplex table is very easy. We pick from the tableau the column corresponding to $\boldsymbol{u}_k$ and subtract the current rhs ($\boldsymbol{u}_0$) from $\boldsymbol{u}_k$.)

Step 2: Selecting the Leaving Variable

Increase t: $t_0 \rightarrow \min\{1, t_B\}$, where t_B corresponds to the value of t, where the next basis change happens.

If $t < 1$

Go to **Step 3**

Else

Define $\boldsymbol{u_0} := \boldsymbol{u_0} + \boldsymbol{\delta}$

If $\sigma > 0$ (the unit k is inefficient)

[4] The sequence of the units to be checked has an effect on the performance of the procedure. In this paper, we do not deal with that problem.

Define $K^I := K^I \cup \{k\}$.

If $\sigma = 0$ and $\boldsymbol{1}^T\boldsymbol{s} \neq 0$ (the unit k is weakly efficient)

Define $K^W := K^W \cup \{k\}$.

If $\sigma = 0$ and $\boldsymbol{1}^T\boldsymbol{s} = 0$ (the unit k is efficient but non-extreme)

Define $K^{EN} := K^{EN} \cup \{k\}$.

Define $K^U := K^U \cap -\{k\}$. Drop the column corresponding to unit k from the matrix **U**. Go to **Step 1**.

Step 3: Selecting the Entering Variable

If the entering variable is not uniquely defined, use the lexicographic rule (2.5) to check the optimality conditions to find an entering variable keeping the basis lexicographically optimal.

If the entering variable is one of the slack variables

Set $t_0 := t_B$, make a basis change and go to **Step 2** (or to **Step 4** in the case the extra efficiency checking is made for non-basic units with unknown status).

Else

Assume that the index of the entering variable h corresponds to unit DMU$_h$ (unit h is efficient extreme).

Define $K^U := K^U \cap -\{h\}$ and $K^{EE} := K^{EE} \cup \{h\}$.

If $h = k$

Define $\boldsymbol{u_0} := \boldsymbol{u_0} + t_0\boldsymbol{\delta}$ and go to **Step 1**.

Else

Set $t_0 := t_B$, make a basis change and go to **Step 2** (or **Step 4**).

Step 4: Checking Inefficient Units

Check all (non-basic) units with status unknown. If all the values are non-negative in the column of the simplex tableau corresponding to unit $i \in K^U$, then:

- unit i is inefficient, if the value in row "σ" is strictly positive,
- unit i is weakly efficient, if the value in row "σ" is zero, but any of the slack-variables $\boldsymbol{s}$ has a strictly positive value,
- otherwise the unit is not efficient extreme. (It locates on an efficient facet defined by the current basis units.)

If any of the units is diagnosed inefficient, weakly efficient or not efficient extreme update the sets K^I, K^W, K^{EN} and K^U accordingly. Drop the columns corresponding to inefficient, weakly efficient and not efficient extreme units from the matrix **U**.

If one of the units diagnosed is DMU$_k$ (= the unit under consideration) or DMU$_k$ is already classified, go to **Step 1**; otherwise go to **Step 2.**

4.2 Illustrative Numerical Example

Our numerical example is illustrated in Figure 1. Assume that we have six units which are evaluated with three outputs and one identical input as shown in Table 1.

Table 1: Illustrative Example

	A	B	C	D	E	F
Output 1	1	0.9	1.5	4	1.5	2.75
Output 2	0	2	3	1	2.5	2
Output 3	3	2.5	1.5	2	1.6	0.45
Input	1	1	1	1	1	1

In the illustration of the numerical example, we use the capital letters A, B, …, F to refer to the units, the input/output-vectors, the indices of the units, and the columns in the simplex-tableau. Correspondingly, we use the terms Input, Output 1, …, Output 3 to refer to slack- and surplus-variables in the same way.

Step 0:

In this example we choose $w_{\text{Output 1}} = w_{\text{Output 2}} = w_{\text{Output3}} = 1$ and $w_{\text{Input}} = 0$.

Start the procedure by investigating the efficiency of unit A. Define the rhs-vector $\boldsymbol{u}_0$ of the problem (3.1): $\boldsymbol{u}_0 := \boldsymbol{u}_{\text{A}}$ and solve it. (We use the symbol $\boldsymbol{u}_0$ to refer to the rhs in the tableau as well.)

An optimal tableau is given in Table 1. Unit A is one of the basic variables in the optimal basis, thus it is efficient extreme. It is the only basic variable associated with a unit. The optimal tableau is not uniquely defined, because A is the only basic variable with value greater than zero. Any of those alternative optimal bases can be used as a starting basis. Because all values in row "σ" are strictly positive, the second objective is not needed.

Table 2: An optimal tableau for unit A

	Output 3	B	C	D	E	F	Input	$\boldsymbol{u}_0$
Output 1	-1	-0.4	-2	-4	-1.9	-4.3	-2	0
Output 2	-1	-2.5	-4.5	-2	-3.9	-4.55	-3	0
A	0	1	1	1	1	1	1	1
σ	1	0.5	1.5	1	1.4	2.55	3	0

Define $\text{K}^{\text{EE}} := \{\text{A}\}$, $\text{K}^{\text{EN}} := \varnothing$, $\text{K}^{\text{W}} := \varnothing$, $\text{K}^{\text{I}} := \varnothing$, and $\text{K}^{\text{U}} := \{\text{B, C, D, E, F}\}$.

Step 1:

Choose unit C next, and define the parameter vector $\boldsymbol{\delta} := \boldsymbol{u}_{\text{C}} - \boldsymbol{u}_{\text{A}}$. (In the tableau, we subtract column $\boldsymbol{u}_0$ from the column C. The problem formulation is displayed in Table 3.

Table 3: The problem formulation for moving from A to C

	Output 3	B	C	D	E	F	Input	u_0	δ
Output 1	-1	-0.4	-2	-4	-1.9	-4.3	-2	0	-2
Output 2	-1	-2.5	-4.5	-2	-3.9	-4.55	-3	0	-4.5
A	0	1	1	1	1	1	1	1	0
σ	1	0.5	1.5	1	1.4	2.55	3	0	1.5

Step 2:

We cannot increase parameter t before we have the basis in which the positive values of the element of the parameter vector correspond to zeroes in the rhs. Select as the basis leaving variable the variable (Output 2) which has the biggest negative value in the parameter vector corresponding to the zero value of the rhs in the same row.

Step 3:

Variable B enters the basis (See, Table 4), and it is diagnosed efficient extreme. Redefine K^U := {C, D, E, F} and K^{EE} := {A, B}.

Table 4: Variable Output 2 is replaced by variable B in the basis

	Output 3	Output 2	C	D	E	F	Input	U_0	δ
Output 1	-0.84	-0.16	-1.28	-3.68	-1.276	-3.572	-1.52	0	-1.28
B	0.4	-0.4	1.8	0.8	1.56	1.82	1.2	0	1.8
A	-0.4	0.4	-0.8	0.2	-0.56	-0.82	-0.2	1	-1.8
σ	0.8	0.2	0.6	0.6	0.62	1.64	2.4	0	0.6

Next we visit **Step 2** and notice that we cannot increase t before we return back to **Step 3** to remove also variable Output 1 out of the basis. Variable D enters the basis; it is diagnosed efficient extreme. The tableau is shown in Table 5. Redefine K^U := {C, E, F} and K^{EE} := {A, B, D}, and return to **Step 2**.

Table 5: Starting the moving from A to C

	Output 3	Output 2	C	Output 1	E	F	Input	u_0	δ
D	0.228	0.043	0.348	-0.272	0.347	0.971	0.413	0.000	0.348
B	0.217	-0.435	1.522	0.217	1.283	1.043	0.870	0.000	1.522
A	-0.446	0.391	-0.870	0.054	-0.629	-1.014	-0.283	1.000	-1.870
σ	0.663	0.174	0.391	0.163	0.412	1.058	2.152	0.000	0.391

Step 2:

The current basis is feasible until t reaches t_B = 0.535. Because $t = t_B$ is less than 1, we set t_0:= 0.535, and go to **Step 3**.

Step 3:

Unit A is the basis leaving variable and unit C is the basis entering variable. Unit C is diagnosed efficient extreme. We redefine K^U := {E, F} and K^{EE} := {A, B, C, D}. Because the entering unit corresponds the unit under consideration ($h = k$ in Step 3) we also need to redefine $\boldsymbol{u}_0$. In this example, we go to **Step 4** to check if any of the non-basic variables can be diagnosed. Redefine $\boldsymbol{u}_0 := \boldsymbol{u}_0 + 0.535^*\boldsymbol{\delta}$ (See, Table 6) and make a basis change.

Table 6: Unit A is replaced by unit C in the basis

	Output 3	Output 2	A	Output 1	E	F	Input	$\boldsymbol{u}_0$	$\boldsymbol{\delta}$
D	0.050	0.200	0.400	-0.250	0.095	0.565	0.300	0.186	-0.400
B	-0.563	0.250	1.750	0.313	0.181	-0.731	0.375	0.814	-1.750
C	0.513	-0.450	-1.150	-0.063	0.724	1.166	0.325	0.000	2.150
σ	0.463	0.350	0.450	0.188	0.129	0.601	2.025	0.209	-0.450

Step 4:

All values in the column corresponding to unit E in Table 6 are strictly positive. We may diagnose unit E inefficient. Thus redefine $K^U := \{F\}$, $K^I := \{E\}$, and $K^{EE} := \{A, B, C, D\}$. Drop column E from the tableau and go to **Step 1.**

Step 1:

We start to study the efficiency of a new unit using the current rhs-vector (See, Table 6). Unit F is the only unit with an unknown status. We formulate the problem for F. The parameter vector is $\boldsymbol{\delta} := \boldsymbol{u}_F - \boldsymbol{u}_0$ (See, Table 7). Go to **Step 2.**

Table 7: The entering variable is not uniquely defined

	Output 3	Output 2	A	Output 1	F	Input	$\boldsymbol{u}_0$	$\boldsymbol{\delta}$
D	0.050	0.200	0.400	-0.250	0.565	0.300	0.186	0.379
B	**-0.563**	0.250	1.750	0.313	**-0.731**	0.375	0.814	-1.545
C	0.513	-0.450	-1.150	-0.063	1.166	0.325	0.000	1.166
σ	0.463	0.350	0.450	0.188	0.601	2.025	0.209	0.392

Step 2:

We may increase t: 0 → 0.527. Update the rhs-vector and go to **Step 3**.

Step 3:

The leaving variable is B, but the entering variable is not uniquely defined. We have to use the lexicographic rule to choose the right one.

Table 8: The current reduced costs for alternative entering variables

	Output 3	Output 2	A	Output 1	F	Input	$\boldsymbol{u}_0$	$\boldsymbol{\delta}$
D	0.050	0.200	0.400	-0.250	0.565	0.300	0.386	0.379
B	**-0.563**	0.250	1.750	0.313	**-0.731**	0.375	0.000	-1.545
C	0.513	-0.450	-1.150	-0.063	1.166	0.325	0.614	1.166
σ	0.463	0.350	0.450	0.188	0.601	2.025	0.416	0.392
$\boldsymbol{1}^T\boldsymbol{s}$	-1	*	*	*	0	*		

The alternative entering variables are Output 3 and F. For the both variables, the ratio is 0.463/-0.563 = 0.601/-0.731 = -0.822. When we use the second objective, we get $(z^2_{Output\ 3} - c^2_{Output\ 3})/y_{B,Output\ 3} = (\boldsymbol{c}^2_B\mathbf{B}^{-1}\boldsymbol{a}_j - c^2_{Output\ 3})/\ y_{B,Output\ 3} =$ 1/-0.563 = 1.778 >

$(z_F^2 - c_F^2)/y_{B,F} = (\boldsymbol{c}^2{}_B\mathbf{B}^{-1}\boldsymbol{a}_j - c_F^2)/\ y_{B,F} = $ 0/-0731 = 0.

Thus the correct entering variable guaranteeing the efficient basis is Output 3.

After the basis change, we get the final tableau (Table 9), which is lexicographically optimal.

Table 9: A lexicographically optimal basis

	B	Output 2	A	Output 1	F	Input	u_0	δ
D	0.089	0.222	0.556	-0.222	0.500	0.333	0.386	0.242
Output 3	-1.778	-0.444	-3.111	-0.556	1.300	-0.667	0.000	2.747
C	0.911	-0.222	0.444	0.222	0.500	0.667	0.614	-0.242
σ	0.822	0.556	1.889	0.444	0.000	2.333	0.416	-0.879
1^Ts	-1.778	*	*	*	1.300	*	*	*

We continue to **Step 4.**

Step 4:

We see that all values in column F are strictly positive. F is inefficient. We update $K^U := \varnothing$, $K^I := \{E, F\}$, and $K^{EE} := \{A, B, C, D\}$. Because F was the unit under consideration, we go to **Step 1** to stop the search.

5 Computational Results

To test the performance of our procedure, we made some computational tests with simulated models, which we received from Prof. Jose Dulá who has also used these models in his own tests. The parameters of the problems are the number of units, the number of inputs/outputs, and the density of the efficient units. The categories for the number of units we used 5000, 10000, 15000, 20000, 25000, and 50000. The number of inputs/outputs was 5, 10, 15, and 20. As the density categories we used 1%, 10%, and 25%.

The test results are reported in Table 10 and Table 11 in each combination of the parameters except some missing results for 50000 units. The tests are run with the PC-computer with one 2.4 Ghz processor. In Table 10, we have reported the procedure in which the efficiency of each unit is checked at a time. No extra information available in the simplex tableau is used to check the efficiency of other units with unknown status, i.e. excluding the use of Step 4. In Table 11, this information is used.

We can see from the Tables that the procedure with Step 4 is more efficient, when the number of the inputs/outputs is 5 in all other categories, and it is also more efficient, when the number of inputs/outputs is 10, the density is 10% and the number of units is less than 20000. In other cases, it is slower. It is understandable, because extra checking is costly, and the benefit of it's use decreases, when the efficiency density and/or the number of inputs/outputs increase.

Table 10: Computing Times (s) when Step 4 is Not Included

Density	# of Alternatives	# of Inputs/Outputs			
		5	10	15	20
1 %	5 000	12	64	176	354
	10 000	60	310	743	1686
	15 000	164	897	2003	3559
	20 000	275	1643	4266	8589
	25 000	588	2946	5309	12342
	50 000	2226	14633	26623	
10 %	5 000	24	86	294	555
	10 000	119	628	1484	2935
	15 000	325	1718	4147	6864
	20 000	691	3590	9154	11425
	25 000	1213	5264	9961	19209
	50 000	5839	22240		
25 %	5 000	34	116	381	675
	10 000	189	844	1895	3264
	15 000	530	1965	4394	8053
	20 000	1126	4503	8245	15756
	25 000	2055	6874	13263	25089
	50 000	9006	30454		

Table 11: Computing Times (s) when Step 4 is Included

Density	# of Alternatives	# of Inputs/Outputs			
		5	10	15	20
1 %	5 000	0.2	25	176	394
	10 000	0.5	169	870	1904
	15 000	1.6	553	2369	4159
	20 000	2.5	1091	5236	8759
	25 000	5.3	2398	7499	14716
	50 000	14.1	10110		
10 %	5 000	1	72	335	623
	10 000	4	604	1762	3288
	15 000	9	1646	4542	7847
	20 000	23	3948	8297	13193
	25 000	37	6141	12384	22225
	50 000	143			
25 %	5 000	2	120	411	776
	10 000	11	938	2155	3588
	15 000	32	2273	4893	9571
	20 000	66	5220	9465	17828
	25 000	107	8591	15471	27857
	50 000	442			

The complexity of the procedure depends on *n*, *p*, *d*. Thus we may assume that the complexity is O($n^a p^b d^c$), where *d* the density of the efficiency units. To find the values for a, *b¸* and *c*. We used the data in the tables above to fit a regression function to the data. We used the following model:

$t = kn^a p^b d^c \varepsilon,$

where ε is an error term assumed normally distributed, and k is also the parameter to be estimated. We took the logarithm of the model and used a linear regression model to estimate the parameters.

For the basic procedure (Table 10), we obtained the following estimates.

Table 12: The estimates of the regression coefficients for the model of the basic procedure ($R^2 = 0.994$)

	Coefficients	*Standard Error*	*Lower 95%*	*Upper 95%*
log(*k*)	-9.044	0.134	-9.312	-8.775
a	2.165	0.035	2.096	2.235
b	2.343	0.030	2.284	2.403
c	0.282	0.014	0.255	0.309

The model fitted very well with the data. The estimates of the parameters are consistent with our expectations. When *p* is rather small, obviously we will find quite quickly the potential basic vectors (efficient units). In case of two input/output-vectors, we may diagnose all efficient units with one iteration. Interestingly, the estimate of the power of the density is low. It means that the density has not so high impact on the performance as we expected.

Instead our model for the computing times of the modified procedure resulted in a little bit strange results (Table 13). The estimate of parameter *a* is about 5. The reason is that the procedure works very well with a small number of the inputs/outputs and the small efficiency density, but the computing time increases very rapidly, when the values of those parameters increase.

Table 13: The estimates of the regression coefficients for the model of the modified procedure ($R^2 = 0.929$)

	Coefficients	*Standard Error*	*Lower 95%*	*Upper 95%*
log(*k*)	-12.162	0.763	-13.690	-10.635
a	5.023	0.198	4.627	5.419
b	2.241	0.170	1.902	2.581
c	0.485	0.077	0.330	0.640

The estimates of parameters of the model were very much the same as the model for the basic procedure, when considered the large models by restricting our considerations into the following values of parameters: when $p = 10$, we picked the models with $n \geq 15000$ and $d \geq 10$, and then we picked all models with $p \geq 15$ (Table 14). It means that the use of extra information is not beneficial, when the number of the units and the efficiency density is big.

Table 14: The estimates of the regression coefficients for the model of the modified procedure, when *n*, *p* and *d* are big ($R^2 = 0.994$)

	Coefficients	*Standard Error*	*Lower 95%*	*Upper 95%*
log(*k*)	-8.722	0.162	-9.051	-8.392
a	2.057	0.070	1.914	2.200
b	2.321	0.031	2.258	2.384
c	0.230	0.014	0.201	0.258

The performance of the modified procedure especially for the small models not included in the previous analysis is interesting. In Table 15, we see log(*k*) is very small indicating a good performance, when the number of inputs/outputs *(p*) is small. However, the time increases rapidly, when *p* increases. When *p* (and *n*) is large the performance of the modified procedure is of the same magnitude as the basic one.

Table 15: The estimates of the regression coefficients for the model of the modified procedure, when *n*, *p* and *d* are small. ($R^2 = 0.978$)

	Coefficients	*Standard Error*	*Lower 95%*	*Upper 95%*
log(*a*)	-15.065	0.598	-16.309	-13.821
b	7.806	0.287	7.209	8.403
c	2.355	0.116	2.113	2.597
d	0.824	0.063	0.694	0.954

6 Concluding Remarks

We have developed a procedure for the classification of efficient and inefficient units in the Data Envelopment Analysis (DEA). This classification is important, because the algorithms currently developed for large-scale problems are based on a two-phase procedure. In the first phase, the efficient units are recognized, and in the second phase those units are used as potential basic vectors in computation of the efficiency scores for inefficient units. Because the proportion of the efficient units in practical problems is usually quite small, the size of the LP-models in the second phase is much smaller than in the first phase.

Our classification procedure is based on the lexicographic parametric programming. The procedure makes it possible to move from unit to unit along an efficient curve. On the way, it is possible to use information available in the simplex tableau to check the efficiency of the other units with status unknown. This modified procedure is very fast for small number of inputs/outputs, but its performance is getting worse rapidly, when the number of inputs/outputs increases. With a large number of inputs/outputs, its performance is of the same magnitude as the basic procedure.

In the future, our purpose is to develop various decomposition techniques, which make it possible to take a full benefit from the good performance of our modified procedure for small number of inputs/outputs and units.

References

Ali, A.I. (1993), "Streamlined Computation for Data Envelopment Analysis",

European Journal of Operational Research 64, 61-67.

Banker, R.D., Charnes, A. and Cooper, W.W. (1984), "Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis", Management Science 30, 1078-1092.

Barr, R.S. and Durchholz, M.L. (1997). "Parallel and Hierarchical Decomposition Approaches for Solving Large-Scale Data Envelopment Analysis Models", Annals of Operations Research 73, 339-372.

Chambers R. G., Y. Chung, and R. Färe (1998), "Profit, Directional Distance Functions, and Nerlovian Efficiency", Journal of Optimization Theory and Applications 98, 351-364.

Charnes, A., Cooper, W.W. and Rhodes, E. (1978), "Measuring Efficiency of Decision Making Units", European Journal of Operational Research 2, 429-444.

Charnes, A., Cooper, W.W. and Rhodes, E. (1979), "Short Communication: Measuring Efficiency of Decision Making Units", European Journal of Operational Research 3, 339.

Charnes, A., Cooper, W., Lewin, A.Y. and Seiford, L.M. (1994**),** Data Envelopment Analysis: Theory, Methodology and Applications, Kluwer Academic Publishers, Norwell.

Dulá, J.H., and Helgason, R.V. (1996), "A New Procedure for Identifying the Frame of the Convex Hull of a Finite Collection of Points in Multidimensional Space", European Journal of Operational Research 92, 352-367.

Dulá, J.H., Helgason, R.V., and Venugopal, N. (1997). "An Algorithm for Identifying the Frame of a Pointed Finite Conical Hull", Journal of Computing 10, 323-330.

Dulá, J. H., and López, F. J. (2002), "Data Envelopment Analysis (DEA) in Massive Data Sets", in Abello, J. , Pardalos, P., and Resende, M. (Eds.): Handbook of Massive Data Sets, Kluwer Academic Publisher, pp. 419-437.

Halme, M., Joro, T., Korhonen, P., Salo, S. and Wallenius J. (1999), "A Value Efficiency Approach to Incorporating Preference Information in Data Envelopment Analysis", Management Science 45, pp. 103-115.

Korhonen, P. and Halme, M. (1996), "Using Lexicographic Parametric Programming for Searching a Nondominated Set in Multiple Objective Linear Programming", Journal of Multi-Criteria Decision Analysis 5, 291-300.

Sawaragi, Y., Nakayama, H. and Tanino, T. (1985), "Theory of Multiobjective Obtimization", Academic Press, Inc.

Steuer, R. E. (1986), Multiple Criteria Optimization: Theory, Computation, and Application, Wiley.