EFFICIENCY EVALUATION BY ANALYZING THE EFFICIENT FRONTIER IN DATA ENVELOPMENT ANALYSIS

Satoshi Washio

Doctoral Program in
Information Science and Engineering
Graduate School of Science and Technology
Niigata University

Contents

Notation

$\mathbb{R}^n$	:	an n-dimensional Euclidean space.
$\operatorname{int} C$	:	the topological interior of a subset C of a topological space.
$V \setminus \{x_0\}$	:	the set removed x_0 from V.
$\|\|x\|\|$	:	the Euclidean norm of x.
$a^\top$	:	the transposed vector of a in $\mathbb{R}^n$.
$\dim S$	:	the dimension of S.
$V(S)$	:	the set of all vertices of a convex polyhedral set S.

Chapter 1

INTRODUCTION

This thesis is concerned with evaluation methods of DMUs (Decision Making Units) in DEA (Data Envelopment Analysis). DEA is a non-parametric analytical method for estimating relative efficiencies of DMUs performing similar tasks that consumes inputs to produce outputs. DEA has been developed as a methodology used for efficiency analysis of DMUs for about thirty years. In order to evaluate the efficiencies of DMUs, the regression analysis has been researched. The regression analysis focuses on setting up a regression line which passes through the center of DMUs and evaluates DMUs based on the line. That is, all DMUs are evaluated by a fixed weight. In contrast, each DMU is evaluated based on a most advantageous weight for itself which is obtained by solving a linear programming problem in DEA. The idea of DEA is to identify best performance DMUs within a set of comparable DMUs and those form an efficient frontier. The regression analysis is an analytical method based on the average, while DEA is a evaluation method with reference to the superior DMUs.

1.1 DEA models

DEA has been proposed by Charnes, Cooper and Rhodes [9] as the CCR model. In the CCR model, DMUs are evaluated under a condition of the constant returns to scale which means that all efficient DMUs can produce up to k outputs by using k inputs. In order to deal with variable returns to scale, Banker, Charnes and Cooper [4] have proposed the BCC model. In addition, the DRS and IRS models have been proposed to deal with decreasing and increasing returns to scale by Fare and Grosskopt [20], and Seiford and Thrall [30], respectively. These models have the properties of radial measure. Two main types of efficiency measures in DEA are radial and non-radial measures. In the radial measure models, an efficiency score of DMU is determined as the reduction scale to become an efficient unit. By reference to some radial measure models, many non-radial measure models have been proposed. In the non-radial measure models, an efficiency score of DMU is determined by using slacks which mean how far apart from an efficient frontier. The additive model proposed by Charnes, Cooper, Golany, Seiford and Stutz [7] is one of the traditional non-radial measure model. This model has the advantage that it is translation invariant (see Ali and Seiford [1], Lovell and Pastor [27], Pastor [29]). By transforming the data using in the additive model by the natural logarithm, the multiplicative model has been formulated [11, 12]. Recently, Tone [39] has proposed the SBM model to consider slacks directly. In 2010, the epsilon-based measure model has been proposed by uniting the radial and the non-radial measure models (see [38]). Moreover, many models have been proposed to cope with practical situations. Banker and Morey [5] have proposed a model including some nondiscretionary variables. Moreover, Sengupta [31] have proposed the stochastic DEA to treat a data uncertainty. Entani, Maeda and Tanaka [19] have proposed the interval DEA by defining an efficiency as an interval to evaluate DMUs realistically.

1.2 Improvements for inefficient DMUs

In DEA, for each DMU, the evaluated value of the efficiency is defined as the optimal value of a linear programming problem. Moreover, for each inefficient DMU, an improvement is obtained simply by solving the problem. Therefore, the other improvement has not been researched exactly until recently. However, it is often difficult to improve the values of inputs and outputs according to the improvement. Because the improvement obtained by the radial measure models improve the only input (or output) values at the same rate. Therefore, Frei and Harker [21] have proposed a least distance projection to the efficient frontier by using the Euclidean norm. Moreover, Takeda and Nishino [34] have proposed the minimal norm problem to the efficient frontier from an inefficient DMU. Recently, for each inefficient DMU, the study of improvements of efficiency is one of the important subjects in DEA. Aparicio, Ruiz and Sirvent [3] have formulated some mixed integer linear programming problems for typical norms to obtain a closest target on the efficient frontier under a certain distance. Lozano and Villa [28] have proposed a gradual efficiency improvement strategy.

In this thesis, we propose an algorithm to calculate a flexible improvement by introducing a policy of the decision maker. In order to obtain improvements of DMUs, we use all equations forming the facets of the efficient frontiers. Therefore, we propose algorithms for constructing the equations forming the efficient frontiers.

1.3 Ranking methods of DMUs

In DEA, each DMU is classified as either inefficient or efficient based on the optimal value of each model. In general, several DMUs are evaluated as efficient and have no inferior-to-superior relationship among them. For example, the departments data in a university investigated by Wong and Beasley [42], six of seven departments were evaluated as efficient units. In practical problems, it is necessary that the decision maker knows the dominance relationships among all DMUs to maximize his profit. In DEA, there are representative analytical methods for ranking DMUs - the sensitivity analysis (see [2, 8, 15, 23]), the assurance region methods (see [14, 35, 36, 37, 44]) and the cross efficiency evaluation (see [18, 22, 32]).

In the sensitivity analysis, DMUs are analyzed based on the change of the efficiency scores by changing the number of DMUs or inputs or outputs. Sometimes, in some traditional DEA models, an optimal solution has zero components. Having zero components means that the inputs or outputs corresponding to the zero components are not completely used to evaluate the DMU. The assurance region method have been proposed to overcome this phenomenon. This method introduces some conditions to input-output variables (for example, the ratio between two input-output variables, magnitude relation , importance condition and so on). In general, each DMU is evaluated by only advantageous weight for itself. In the cross efficiency evaluation, each DMU gets many efficiency scores by using optimal solutions of the CCR model for all DMUs. Since the CCR model often has many optimal solutions for each efficient DMU, we need to decide an optimal solution uniquely for each DMU. Sexton, Silkman and Hogan [32] have formulated a problem to obtain a weight minimizing the sum of cross efficiency scores of the other DMUs. However, it is difficult to solve the mathematical programming problem formulated for this purpose. Therefore, one of the most commonly used secondary goal approach to decide an optimal solution uniquely is suggested by Doyle and Green [18], which is called the aggressive formulation and is formulated as linear programming problems. Recently, Wu, Liang, Zha and Yang [43] have proposed a cross efficiency evaluation based on rank priority. Moreover, Wang and Chin [40] have proposed a neutral model for cross efficiency which seeks a common set of weights for all DMUs.

In this thesis, we propose five types of methods to evaluate DMUs by utilizing the equations forming the facets of the efficient frontier. By this approach, we obtain the same scores as the two kinds of existing approaches without solving linear programming problems. Moreover, we improve the aggressive formulation to obtain a closer cross efficiency score to the aim minimizing the sum of cross efficiency scores of the other DMUs than the traditional formulations.

1.4 Organization of this thesis

This thesis is organized as follows. In Chapter 2, we provide some mathematical preliminaries which will be used in this thesis. In Chapter 3, we introduce some basic DEA models. In Chapter 4, we introduce the previous researches formulated as mixed integer linear programming problems for calculating the equations forming the facets of the efficient frontier and improvements. In contrast, we propose different approaches to calculate four kinds of improvements for inefficient DMUs in the CCR model. In order to calculate the improvements, we use all equations forming the facets of the efficient frontier. Therefore, we propose three types of algorithms to obtain them. Moreover, we show a numerical experiment to compare the improvements proposed in this chapter. In Chapter 5, we introduce some of the previous researches with respect to the cross efficiency evaluation. By analyzing the efficient frontier of the CCR model, we propose five kinds of evaluation methods having the dominance relationships for all DMUs. In the first and the second measures, the weighted sum of the scores calculated based on the equations forming the facets of the efficient frontier is calculated by deciding a weight of each facet. In the other measures, we calculate the cross efficiency scores by using some equations forming the facets of the efficient frontier.

Chapter 2

PRELIMINARIES

In this chapter, we give some mathematical preliminaries which will be used in this thesis.

2.1 Basic definitions and theorems in convex analysis

In order to construct all equations forming the facets of the efficient frontiers, we utilize convex optimization techniques. Therefore, we show several definitions and lemmas in convex analysis.

Definition 2.1.1. Let E be a nonempty subset in $\mathbb{R}^n$. Then, E^* is called the *polar set* of E if it is defined as follows.

$$E^* := \{y \in \mathbb{R}^n : y^\top x \leq 1 \text{ for all } x \in E\}.$$

Definition 2.1.2. Let E be a nonempty subset in $\mathbb{R}^n$. Then, $\mathrm{co}(E)$ is said to be the *convex hull* of E if $\mathrm{co}(E)$ is defined as follows.

$$\mathrm{co}(E) := \left\{ x \in \mathbb{R}^n : x = \sum_{j=1}^{m} \lambda_j x(j),\ \sum_{j=1}^{m} \lambda_j = 1,\ x(j) \in E,\ \lambda_j \geq 0,\ j = 1, \ldots, m \right\}.$$

Definition 2.1.3. Let E be a nonempty subset in $\mathbb{R}^n$. Then, $\mathrm{conic}\, E$ is called the *conical hull* of E if it is defined as follows.

$$\mathrm{conic}\, E := \left\{ x \in \mathbb{R}^n : x = \sum_{j=1}^{m} \lambda_j x(j),\ x(j) \in E,\ \lambda_j \geq 0,\ j = 1, \ldots, m \right\}.$$

Definition 2.1.4. Let E be a nonempty subset in $\mathbb{R}^n$. If E is defined as $E = \{x \in \mathbb{R}^n : Ax \leq b\}$ for some $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, then, E is called the *polyhedral set.* In particular, if E is bounded, then, E is called the *polytope.*

Definition 2.1.5. Let E be a polytope in $\mathbb{R}^n$ satisfying $\dim E = n$. Then, $F := E \cap \{x \in \mathbb{R}^n : a^\top x = b\}$ is called the *facet* of E if $a^\top x \leq b$ for each $x \in E$ and $\dim F = n - 1$.

Definition 2.1.6. Let E be a nonempty closed convex subset in $\mathbb{R}^n$. A nonzero vector d in $\mathbb{R}^n$ is called a *direction* of E if $x + \lambda d \in E$ for each $x \in E$ and $\lambda \geq 0$. Two directions d_1 and d_2 of E are called *distinct* if $d_1 \neq \alpha d_2$ for each $\alpha > 0$. A direction d of E is called an *extreme direction* if it cannot be written as a positive linear combination of two distinct directions, that is, if $d = \lambda_1 d_1 + \lambda_2 d_2$ for some λ_1, $\lambda_2 > 0$, then $d_1 = \alpha d_2$ for some $\alpha > 0$.

Definition 2.1.7. Let E be a nonempty closed convex subset in $\mathbb{R}^n$. Then, E^+ is called the *recession cone* of E if it is defined as follows.

$$E^+ := \{d : x + rd \in E \text{ for all } x \in E, r \geq 0\}.$$

Lemma 2.1.1. *Let E be a nonempty set in $\mathbb{R}^n$. Then E^* is a closed convex set.*

Proof. By Definition 2.1.1, $E^* = \bigcap_{x \in E}\{y : x^\top y \leq 1\}$. For all $x \in E$, $\{y : x^\top y \leq 1\}$ is closed convex set. Therefore, E^* is a closed convex set. □

Lemma 2.1.2. (Konno, Thach and Tuy [26], Proposition 2.6) *Let E be a nonempty closed convex set in $\mathbb{R}^n$ and $0 \in E$. Then $E^{**} = E$.*

Lemma 2.1.3. *Let E be a polytope in $\mathbb{R}^n$. Then, $E^* = (V(E))^*$.*

Proof. Let $V(E) := \{a^1, \ldots, a^m\}$. Since E is a polytope, E is expressed as follows [6].

$$E = \left\{x : x = \sum_{i=1}^{m} \lambda_i a^i,\ \sum_{i=1}^{m} \lambda_i = 1,\ \lambda_i \geq 0, i = 1, \ldots, m\right\}. \tag{2.1}$$

Obviously, $E \supset V(E)$. From the characteristic of the polar set, $E^* \subset (V(E))^*$. Hence, we shall show that $E^* \supset (V(E))^*$. Let $y \in (V(E))^*$. Then, for each $i = 1, \ldots m$, ${a^i}^\top y \leq 1$. By (2.1), for each $x \in E$, there exists $\lambda^x \in \mathbb{R}^m$ such that $x = \sum_{i=1}^m \lambda_i^x a^i$, $\sum_{i=1}^m \lambda_i^x = 1$, $\lambda_i^x \geq 0$ $i = 1, \ldots, m$. Then, $x^\top y = (\sum_{i=1}^m \lambda_i^x a^i)^\top y = \sum_{i=1}^m \lambda_i^x {a^i}^\top y \leq \sum_{i=1}^m \lambda_i^x = 1$. Therefore, $y \in E^*$. Consequently, $E^* = (V(E))^*$. □

Lemma 2.1.4. (Jonathan and Adrian [25]) *Let E be a nonempty subset in $\mathbb{R}^n$. Then E is bounded if and only if $0 \in \operatorname{int} E^*$.*

Lemma 2.1.5. *Let E be a polytope in $\mathbb{R}^n$ and $0 \in \operatorname{int} E$. Then, $E = (V(E^*))^*$.*

Proof. From Lemma 2.1.2, $E = E^{**}$. By Lemmas 2.1.3 and 2.1.4, E^* is a polytope if E is a polytope satisfying $0 \in \operatorname{int} E$. From Lemma 2.1.3, $E = E^{**} = (V(E^*))^*$. □

Lemma 2.1.6. *Assume that a polytope $E \subset \mathbb{R}^n$ satisfies $0 \in \operatorname{int} E$. Then, for each $a \in V(E)$, $\dim(E^* \cap \{x \in \mathbb{R}^n : a^\top x = 1\}) = n - 1$.*

Proof. Since $0 \in \operatorname{int} E$, $\dim E = n$. From the boundedness of E and Lemma 2.1.4, $0 \in \operatorname{int} E^*$. This implies that $\dim E^* = n$. Moreover, since $0 \in \operatorname{int} E$ and $a \in V(E) \subset \operatorname{bd} E$, $a \neq 0$ and hence $\dim(E^* \cap \{x : a^\top x = 1\}) \leq n - 1$. Furthermore, since E^* is a polytope and $a \in E$,

$$E^* = \operatorname{co}(V(E^*)) \subset \{x : a^\top x \leq 1\} \tag{2.2}$$

In order to obtain a contradiction, we suppose that $l := \dim(E^* \cap \{x : a^\top x = 1\}) \leq n - 2$. Then, by (2.2), there exists $b^1, \ldots, b^{l+1} \in (V(E^*) \cap \{x : a^\top x = 1\})$ such that $b^1, \ldots, b^{l+1}$ are affine independent. Then, $\dim\{b^1, \ldots, b^{l+1}\} \leq n-1$. Therefore, there exists $b \in \mathbb{R}^n \setminus \{0\}$ such that $b^\top b^i = 0\,(i = 1, \ldots, l+1)$. We note that $v^\top a < 1$ for each $v \in V(E^*) \setminus \{b^1, \ldots, b^{l+1}\}$. Now, for each $v \in V(E^*) \setminus \{b^1, \ldots, b^{l+1}\}$, let

$$\alpha^v := \begin{cases} 1 & \text{if } v^\top b = 0, \\ \dfrac{1 - v^\top a}{|v^\top b|} & \text{if } v^\top b \neq 0. \end{cases}$$

Then, by setting $\bar{\alpha} := \min\{\alpha^v : v \in V(E^*) \setminus \{b^1, \ldots, b^{l+1}\}\}$, we have $v^\top(a \pm \bar{\alpha} b) = v^\top a \pm \bar{\alpha} v^\top b \leq v^\top a + \bar{\alpha}|v^\top b| \leq 1$ for each $v \in V(E^*) \setminus \{b^1, \ldots, b^{l+1}\}$. Moreover, for each $b^i\,(i = 1, \ldots, l+1)$, $b^{i\top}(a \pm \bar{\alpha} b) = b^{i\top} a \pm \bar{\alpha}(b^{i\top} b) = b^{i\top} a = 1$. This implies that $a - \bar{\alpha} b,\ a + \bar{\alpha} b \in (V(E^*))^* = E$. Since, $a = \frac{1}{2}(a - \bar{\alpha} b + a + \bar{\alpha} b)$, this contradicts $v \in V(E)$. Consequently, $\dim(E^* \cap \{x : a^\top x = 1\}) = n - 1$. □

Chapter 3

BASIC DEA MODELS

In this chapter, we introduce some basic DEA models. Through this thesis, n denotes the number of DMUs. Each DMU consumes m different inputs to produce s different outputs. For each $j \in \{1, \dots, n\}$, DMU(j) has an input vector $x(j) := (x(j)_1, \dots, x(j)_m)^\top$ and an output vector $y(j) := (y(j)_1, \dots, y(j)_s)^\top$. Moreover, we assume the following conditions.

(A1) $x(j) > 0$, $y(j) > 0$ for each $j \in \{1, \dots, n\}$.

(A2) $(x(j_1)^\top, y(j_1)^\top)^\top \neq (x(j_2)^\top, y(j_2)^\top)^\top$ for each $j_1, j_2 \in \{1, \dots, n\}$ $(j_1 \neq j_2)$.

(A3) $n > m + s$.

(A4) $\dim(\mathrm{co}(\{(x(1), y(1)), \dots, (x(n), y(n))\})) = m + s$.

Almost all DEA models are formulated under Assumption (A1). Assumptions (A2), (A3) and (A4) are necessary to execute an algorithm to calculate all equations forming the efficient frontier. However, they are satisfied for almost practical problems. Assumption (A4) means that the convex hull of all DMUs has an interior point.

3.1 CCR model

The CCR model formulated by Charnes, Cooper and Rhodes [9] evaluates the ratio between weighted sums of inputs and outputs. The CCR model provides for constant returns to scale (CRS). Therefore, some researchers call the CCR model the CRS model. In order to calculate the efficiency score of DMU(k) $(1 \leq k \leq n)$, the CCR model is formulated as follows:

$$(\mathrm{CCR}(k)) \begin{cases} \text{maximize} & \dfrac{u^{\top}y(k)}{v^{\top}x(k)} \\ \text{subject to} & \dfrac{u^{\top}y(j)}{v^{\top}x(j)} \leq 1,\ j = 1, ..., n, \\ & u_r \geq 0,\ r = 1, ..., s, \\ & v_i \geq 0,\ i = 1, ..., m. \end{cases}$$

Since Problem (CCR(k)) is a fractional programming problem, it is hard to solve Problem (CCR(k)). Therefore, we transform Problem (CCR(k)) into a linear programming problem by setting the denominator of the objective function equals 1:

$$(\mathrm{CCRLP}(k)) \begin{cases} \text{maximize} & u^{\top}y(k) \\ \text{subject to} & v^{\top}x(k) = 1, \\ & u^{\top}y(j) - v^{\top}x(j) \leq 0,\ j = 1, \ldots, n, \\ & u_r \geq 0,\ r = 1, ..., s, \\ & v_i \geq 0,\ i = 1, ..., m. \end{cases}$$

Moreover, we consider the dual problem which is defined as a linear programming problem as follows:

$$(\mathrm{CCRD}(k)) \begin{cases} \text{minimize} & \theta \\ \text{subject to} & \theta x(k)_i - \sum\limits_{j=1}^{n} \lambda_j x(j)_i \geq 0,\ i = 1, \ldots, m, & (3.1) \\ & \sum\limits_{j=1}^{n} \lambda_j y(j)_r - y(k)_r \geq 0,\ r = 1, \ldots, s, & (3.2) \\ & \lambda_j \geq 0,\ j = 1, \ldots, n, & (3.3) \\ & \theta \in \mathbb{R}. \end{cases}$$

Let $\theta^*_{\mathrm{CCR}}(k)$ denote the optimal value of (CCRD(k)). By (3.2) and (3.3), we have that $(\lambda_1, \ldots, \lambda_n) \neq (0, \ldots, 0)$ and hence $\lambda_{\hat{j}} > 0$ for some $\hat{j} \in \{1, \ldots, n\}$. Then, from (3.1), we have $0 \leq \theta^*_{\mathrm{CCR}}(k)x(k)_i - \sum_{j=1}^{n} \lambda_j x(j)_i \leq \theta^*_{\mathrm{CCR}}(k)x(k)_i - \lambda_{\hat{j}} x(\hat{j})_i$. This implies that $\theta^*_{\mathrm{CCR}}(k) > 0$. Moreover, we note that (λ', θ') is a feasible solution of (CCRD(k)), if $\theta' = 1, \lambda'_k = 1$ and $\lambda'_j = 0$ for each $j \in \{1, \ldots, n\} \setminus \{k\}$. Therefore, $0 < \theta^*_{\mathrm{CCR}}(k) \leq 1$. By using the optimal value $\theta^*_{\mathrm{CCR}}(k)$ of (CCRD(k)), the efficiency of DMU(k) for the CCR model is defined as follows:

Definition 3.1.1. If $\theta^*_{\mathrm{CCR}}(k) = 1$ then DMU(k) is said to be CCR-*efficient*. Otherwise, DMU(k) is said to be CCR-*inefficient*.

Sometimes, there exists i (or r) such that $v_i = 0$ (or $u_r = 0$). This means that the i (or r)th input (output) is not completely used to evaluate DMU(k). In order to resolve this shortage, Charnes, Cooper and Rhodes have modified the CCR model by introducing a positive lower limit ($\varepsilon > 0$) in [10]. Then the constraint conditions of Problems (CCR(k)) and (CCRLP(k)) are replaced as follows:

$$\begin{array}{l} v_i \geq 0,\, i = 1, \ldots, m, \\ u_r \geq 0,\, r = 1, \ldots, s. \end{array} \Rightarrow \begin{array}{l} v_i \geq \varepsilon,\, i = 1, \ldots, m, \\ u_r \geq \varepsilon,\, r = 1, \ldots, s. \end{array}$$

The dual problem of Problem (CCRLP(k)) is formulated as follows:

$$(\text{CCRD}\varepsilon(k)) \left\{ \begin{array}{ll} \text{minimize} & \theta - \varepsilon \left(\sum_{i=1}^{m} s_{ix} + \sum_{r=1}^{s} s_{ry} \right) \\ \text{subject to} & \theta x(k)_i - \sum_{j=1}^{n} \lambda_j x(j)_i - s_{ix} = 0,\, i = 1, \ldots, m, \\ & \sum_{j=1}^{n} \lambda_j y(j)_r - y(k)_r - s_{ry} = 0,\, r = 1, \ldots, s, \\ & \lambda_j \geq 0,\, j = 1, \ldots, n, \\ & s_{ix} \geq 0,\, i = 1, \ldots, m, \\ & s_{ry} \geq 0,\, r = 1, \ldots, s, \\ & \theta \in \mathbb{R}. \end{array} \right.$$

By using an optimal solution $(\theta^*_{\text{CCR}}(k), s^*_x, s^*_y)$ of Problem (CCRD$\varepsilon(k)$), the efficiency of DMU(k) for the CCR model is more strictly evaluated.

Definition 3.1.2. If $\theta^*_{\text{CCR}}(k) = 1$ and $(s^*_x, s^*_y) = (0, 0)$ then DMU(k) is said to be CCR-*Pareto-efficient*. If $\theta^*_{\text{CCR}}(k) = 1$ and $(s^*_x, s^*_y) \neq (0, 0)$ then DMU(k) is said to be CCR-*weakly-efficient*. Otherwise, DMU(k) is said to be CCR-*inefficient*.

The presence of an optimal positive slack for some input or output means that the input can be decreased or the output can be increased in T_{CCR} under the condition that the other input and output values are fixed. Therefore, DMU(k) satisfying $\theta^*_{\text{CCR}}(k) = 1$ and $(s^*_x, s^*_y) \neq (0, 0)$ is evaluated as weakly-efficient. Let T_{CCR} be the production possibility set (PPS) of the CCR model defined in [9] as follows:

$$T_{\text{CCR}} := \left\{ (x, y) : x \geq \sum_{j=1}^{n} \lambda_j x(j),\ 0 \leq y \leq \sum_{j=1}^{n} \lambda_j y(j) \text{ for some } \lambda \geq 0 \right\}.$$

By the definitions of T_{CCR} and conical hull, T_{CCR} is represented as follows.

$$T_{\text{CCR}} = (\text{conic}\,\{(x(1), y(1)), \ldots, (x(n), y(n))\} + (\mathbb{R}^m_+ \times \mathbb{R}^s_-)) \cap (\mathbb{R}^m \times \mathbb{R}^s_+).$$

where for a natural number n, $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x(i) \geq 0, i = 1, \ldots, n\}$ and $\mathbb{R}^n_- := \{x \in \mathbb{R}^n : x(i) \leq 0, i = 1, \ldots, n\}$. Obviously, T_{CCR} is a closed convex set. Let F_{CCR} be the efficient forntier of the CCR model which is the emvelope formed by all CCR-efficient DMUs, that is, $F_{\text{CCR}} = \{(x, y) \in (\mathbb{R}^m_+ \times \mathbb{R}^s_+) : \theta^*_{\text{CCR}}((x, y)) = 1\}$.

3.2 BCC model

The BCC model formulated by Banker, Charnes and Cooper [4] has the feasible set defined by adding an equality condition to the constraint conditions of the CCR model. Moreover, the BCC model can classify efficient DMUs into three types of the returns to scale (RTS). To evaluate the efficiency of DMU(k) ($1 \leq k \leq n$), the BCC model is formulated as follows:

$$(\text{BCCLP}(k)) \left\{ \begin{array}{ll} \text{minimize} & \theta \\ \text{subject to} & \theta x(k)_i - \sum_{j=1}^{n} \lambda_j x(j)_i \geq 0,\ i = 1, \ldots, m, \\ & \sum_{j=1}^{n} \lambda_j y(j)_r - y(k)_r \geq 0,\ r = 1, \ldots, s, \\ & \sum_{j=1}^{n} \lambda_j = 1, \\ & \theta \in \mathbb{R},\ \lambda_j \geq 0,\ j = 1, \ldots, n. \end{array} \right.$$

Let $\theta^*_{\text{BCC}}(k)$ denotes the optimal value of Problem (BCCLP(k)). From the definition of the constraint conditions of (BCCLP(k)), it is obvious that $0 < \theta^*_{\text{BCC}}(k) \leq 1$. By using the optimal value $\theta^*_{\text{BCC}}(k)$ of (BCCLP(k)), the efficiency of DMU(k) for (BCCLP(k)) is defined in [4] as follows:

Definition 3.2.1. DMU(k) is said to be BCC-*efficient* if $\theta^*_{\text{BCC}}(k)$=1. Otherwise, DMU(k) is said to be BCC-*inefficient.*

Let T_{BCC} be the PPS of the BCC model as follows:

$$T_{\text{BCC}} := \left\{ (x, y) : x \geq \sum_{j=1}^{n} \lambda_j x(j),\ 0 \leq y \leq \sum_{j=1}^{n} \lambda_j y(j),\ \sum_{j=1}^{n} \lambda_j = 1 \text{ for some } \lambda \geq 0 \right\}.$$

By the definitions of T_{BCC} and convex hull, T_{BCC} is represented as follows:

$$T_{\text{BCC}} = (\text{co}\,(\{(x(1), y(1)), \ldots, (x(n), y(n))\} + (\mathbb{R}^m_+ \times \mathbb{R}^s_-))) \cap (\mathbb{R}^m \times \mathbb{R}^s_+).$$

Obviously, T_{BCC} is a closed convex set. Let F_{BCC} be the efficient forntier of the BCC model which is the emvelope formed by all BCC-efficient DMUs, that is, $F_{\mathrm{BCC}} = \{(x,y) \in (\mathbb{R}^m_+ \times \mathbb{R}^s_+) : \theta^*_{\mathrm{BCC}}((x,y)) = 1\}$.

The RTS expresses a type of the efficiency by the change of the scale about the activity of DMUs. Hence, in the BCC model, there exist three types of the RTS : the increasing RTS, the decreasing RTS and the constant RTS. The increasing and decreasing RTS improve the efficiency by expanding and contracting the scale, respectively. Moreover, the constant RTS means that it is desirable to maintain the present scale.

The following is the dual problem of (BCCLP(k)) ($1 \leq k \leq n$):

$$
(\mathrm{BCCD}(k)) \begin{cases} \text{maximize} & u^\top y(k) - \alpha \\ \text{subject to} & v^\top x(k) = 1, \\ & u^\top y(j) - v^\top x(j) - \alpha \leq 0,\ j = 1, \ldots, n, \\ & u_r \geq 0,\ r = 1, \ldots, s, \\ & v_i \geq 0,\ i = 1, \ldots, m. \end{cases}
$$

Let us set α_* and α^* as follows:

$$
\begin{aligned} \alpha_* := \min\{\alpha : v^\top x(k) = 1,\ & u^\top y(j) - v^\top x(j) - \alpha \leq 0,\ j = 1, \ldots, n, \\ & u_r \geq 0,\ r = 1, \ldots, s,\ v_i \geq 0,\ i = 1, \ldots, m\}. \end{aligned}
$$

$$
\begin{aligned} \alpha^* := \max\{\alpha : v^\top x(k) = 1,\ & u^\top y(j) - v^\top x(j) - \alpha \leq 0,\ j = 1, \ldots, n, \\ & u_r \geq 0,\ r = 1, \ldots, s,\ v_i \geq 0,\ i = 1, \ldots, m\}. \end{aligned}
$$

Then, the RTS is classified as follows (see [17]).

(i) DMU(k) is said to be the increasing RTS if $\alpha_* < \alpha^* \leq 0$ or $\alpha_* = \alpha^* < 0$,

(ii) DMU(k) is said to be the decreasing RTS if $0 \leq \alpha_* < \alpha^*$ or $0 < \alpha_* = \alpha^*$,

(iii) DMU(k) is said to be the constant RTS if $\alpha_* < 0 < \alpha^*$ or $\alpha_* = \alpha^* = 0$,

where $1 \leq k \leq n$.

3.3 GRS model

Similarly, the IRS and DRS models formulated by Seiford and Thrall [30], Fare and Grosskopf [20], respectively, have the feasible set defined by adding an inequality condition to the constraint conditions of the CCR model.

The GRS model unifies the CCR, BCC, IRS and DRS models by introducing an intensity vector λ. In order to calculate the efficiency score of DMU(k) $(1 \leq k \leq n)$, the GRS model is formulated as follows:

$$
(\mathrm{GRS}(k)) \left\{ \begin{array}{ll}
\text{minimize} & \theta \\
\text{subject to} & \theta x(k)_i - \sum_{j=1}^{n} \lambda_j x(j)_i \geq 0 \; i = 1, \ldots, m, \\
& \sum_{j=1}^{n} \lambda_j y(j)_r - y(k)_r \geq 0 \; r = 1, \ldots, s, \\
& L \leq \sum_{j=1}^{n} \lambda_j \leq U, \\
& \theta \in \mathbb{R}, \; \lambda_j \geq 0 \; j = 1, \ldots, n,
\end{array} \right.
$$

where $L \leq 1$ and $U \geq 1$. If $L = 0$ and $U = \infty$, then the model is equivalent to the CCR model. Also, if $L = U = 1$, then the model is the same as the BCC model. If $L = 1$ and $U = \infty$, then the model equals the IRS model. Further, if $L = 0$ and $U = 1$, then the model is the DRS model. Let F_{IRS} and F_{DRS} be the efficient forntiers of the IRS and the DRS models, respectively. Let $\theta^*_{\mathrm{GRS}}(k)$ be the optimal value of $(\mathrm{GRS}(k))$. From the definition of the constraint conditions of $(\mathrm{GRS}(k))$, it is obvious that $0 < \theta^*_{\mathrm{GRS}}(k) \leq 1$. By using the optimal value $\theta^*_{\mathrm{GRS}}(k)$ of $(\mathrm{GRS}(k))$, the efficiency of DMU(k) for $(\mathrm{GRS}(k))$ is defined as follows.

Definition 3.3.1. DMU(k) is said to be GRS-*efficient* if $\theta^*_{\mathrm{GRS}}(k)$=1. Otherwise, DMU$(k)$ is said to be GRS-*inefficient.*

Then, the PPS of the GRS model is defined as follows.

$$
T_{\mathrm{GRS}(L,U)} := \left\{ (x, y) : x \geq \sum_{j=1}^{n} \lambda_j x(j), 0 \leq y \leq \sum_{j=1}^{n} \lambda_j y(j), \; \exists \lambda \in \Lambda(L, U) \right\},
$$

$$
\Lambda(L, U) := \left\{ \lambda \in \mathbb{R}^n : L \leq \sum_{j=1}^{n} \lambda_j \leq U, \; \lambda \geq 0 \right\}.
$$

It is clear that $\Lambda(L, U)$ is a closed convex set for each $L \leq 1$ and $U \geq 1$. Moreover, the following theorem holds.

Theorem 3.3.1. *For each $L \leq 1$ and $U \geq 1$, $T_{\mathrm{GRS}(L,U)}$ is a closed convex set.*

Proof. First, we shall show that $\mathrm{T}(L, U)$ is convex. For each $(x^1, y^1), (x^2, y^2) \in \mathrm{T}(L, U)$, there exist $\lambda^1, \lambda^2 \in \Lambda(L, U)$ such that $x^1 \geq \sum_{j=1}^{n} \lambda^1_j x(j)$, $0 \leq y^1 \leq$

$\sum_{j=1}^n \lambda_j^1 y(j)$, $x^2 \geq \sum_{j=1}^n \lambda_j^2 x(j)$ and $0 \leq y^2 \leq \sum_{j=1}^n \lambda_j^2 y(j)$. For each $0 \leq \alpha \leq 1$, $\alpha\lambda^1 + (1-\alpha)\lambda^2 \in \Lambda(L, U)$. Moreover, $\alpha x^1 + (1-\alpha)x^2 \geq \sum_{j=1}^n (\alpha\lambda_j^1 + (1-\alpha)\lambda_j^2)x(j)$ and $0 \leq \alpha y^1 + (1-\alpha)y^2 \leq \sum_{j=1}^n (\alpha\lambda_j^1 + (1-\alpha)\lambda_j^2)y(j)$. Therefore, $\mathrm{T}(L, U)$ is a convex set.

Second, we shall show that $\mathrm{T}(L, U)$ is closed. Let $\{(x^k, y^k)\} \subset \mathrm{T}(L, U)$ satisfy $(x^k, y^k) \to (\bar{x}, \bar{y})$ as $k \to \infty$. Let $\epsilon > 0$. Since $x^k \to \bar{x}$ as $k \to \infty$, there exists $l \in \mathbb{N}$ such that $||x^k|| \leq ||\bar{x}|| + \epsilon$ for each $k > l$. Let $\delta := \max\{||\bar{x}|| + \epsilon, \max\{||x^k|| : k = 1, \ldots, l\}$. Then, $||x^k|| \leq \delta$ for each $k \in \mathbb{N}$. Since $x(j) > 0$ for each $j = 1, \ldots, n$, $\delta' := \min\{x(j)_i : i = 1, \ldots, m, j = 1, \ldots, n\} > 0$. For each $k \in \mathbb{N}$, $\alpha \in \{\alpha \in \mathbb{R}^n : \sum_{j=1}^n \alpha_j = 1, \alpha_j \geq 0\, j = 1, \ldots, n\}$, we have $x^k \leq \sum_{j=1}^n \alpha_j \frac{\delta}{\delta'} x(j)$. Hence, $\lambda_j \leq \frac{\delta}{\delta'}\alpha_j \leq \frac{\delta}{\delta'}$. For each $k \in \mathbb{N}$, there exists $\lambda^k \in \Lambda(L, U) \cap \{\lambda \in \mathbb{R}^n : 0 \leq \lambda_j \leq \frac{\delta}{\delta'}, j = 1, \ldots, n\}$ such that $x^k \geq \sum_{j=1}^n \lambda_j^k x(j)$, $0 \leq y^k \leq \sum_{j=1}^n \lambda_j^k y(j)$. Since $\{\lambda \in \mathbb{R}^n : 0 \leq \lambda_j \leq \frac{\delta}{\delta'}, j = 1, \ldots, n\}$ is compact, without loss of generality, we can assume that $\lambda^k \to \bar{\lambda}$ as $k \to \infty$. Then, from the closeness of $\Lambda(L, U)$, $\bar{\lambda} \in \Lambda(L, U)$. Hence, $\bar{x} = \lim_{k\to\infty} x^k \leq \lim_{k\to\infty} \sum_{j=1}^n \lambda_j^k x(j) = \sum_{j=1}^n \bar{\lambda}_j x(j)$, $0 \leq \bar{y} = \lim_{k\to\infty} y^k \leq \lim_{k\to\infty} \sum_{j=1}^n \lambda_j^k y(j) = \sum_{j=1}^n \bar{\lambda}_j y(j)$. Therefore $(\bar{x}, \bar{y}) \in \mathrm{T}(L, U)$. Hence, $\mathrm{T}(L, U)$ is closed. Consequently, $\mathrm{T}(L, U)$ is a closed convex set. □

Theorem 3.3.2. *Let* $(x, y) \in F_{\mathrm{IRS}} \cap F_{\mathrm{BCC}}$*,* $L \leq 1$ *and* $U \geq 1$*. Then,* $L(x, y) \in F_{\mathrm{GRS}(L,U)}$*.*

Proof. We show that $L(x, y) \in T_{\mathrm{GRS(L,U)}}$. Let $\bar{\lambda} := L\lambda^*$. Then, $\sum_{j=1}^n \bar{\lambda}_j = \sum_{j=1}^n L\lambda_j^* = L\sum_{j=1}^n \lambda_j^* = L$. Therefore, $L(x, y) \in T_{\mathrm{GRS(L,U)}}$. Let (θ^*, λ^*) be an optimal solution of the following problem.

$$\text{(IRS)}\begin{cases} \text{minimize} & \theta \\ \text{subject to} & \theta x(i) - \sum_{j=1}^{n} \lambda_j x(j)_i \geq 0 \ i = 1, \ldots, m, \\ & \sum_{j=1}^{n} \lambda_j y_{rj} - y_r \geq 0 \ r = 1, \ldots, s, \\ & \sum_{j=1}^{n} \lambda_j \geq 1, \\ & \theta \in \mathbb{R}, \ \lambda_j \geq 0 \ j = 1, \ldots, n. \end{cases}$$

Then, $\theta^* = 1$ and $\sum_{j=1}^n \lambda_j^* \geq 1$. In order to obtain a contradiction, we suppose that $L(x, y) \notin F_{\mathrm{GRS(L,U)}}$. Then, there exist $\theta' < 1$ and λ' satisfying the following conditions.

$$\begin{cases} \theta' Lx(i) - \sum_{j=1}^{n} \lambda_j' x(j)_i \geq 0 \; i = 1, \ldots, m, \\ \sum_{j=1}^{n} \lambda_j' y_{rj} - Ly_r \geq 0 \; r = 1, \ldots, s, \\ L \leq \sum_{j=1}^{n} \lambda_j' \leq U, \\ \lambda_j' \geq 0 \; j = 1, \ldots, n. \end{cases}$$

Let $\bar{\lambda} := \frac{1}{L}\lambda'$. Then, $\sum_{j=1}^{n} \bar{\lambda}_j = \sum_{j=1}^{n} \frac{1}{L}\lambda_j' = \frac{1}{L}\sum_{j=1}^{n} \lambda_j' \geq 1$. Moreover, $\theta' x(i) \geq \sum_{j=1}^{n} \frac{1}{L}\lambda_j' x(j)_i = \sum_{j=1}^{n} \bar{\lambda}_j x(j)_i$ for each $i = 1 \ldots, m$ and $y_r \leq \sum_{j=1}^{n} \frac{1}{L}\lambda_j' y_{rj} = \sum_{j=1}^{n} \bar{\lambda}_j y_{rj}$ for each $r = 1, \ldots, s$. Hence, $(\theta', \bar{\lambda})$ be an feasible solution of Problem (IRS). This contradicts the optimality of (θ^*, λ^*) for Problem (IRS). Therefore, $L(x, y) \in F_{\mathrm{GRS(L,U)}}$. □

Theorem 3.3.3. *Let $(x, y) \in F_{\mathrm{DRS}} \cap F_{\mathrm{BCC}}$, $L \leq 1$ and $U \geq 1$. Then, $U(x, y) \in F_{\mathrm{GRS(L,U)}}$.*

Proof. We can complete the proof in a way similar to Theorem 3.3.2. □

Theorem 3.3.4. *Let $(x, y) \in F_{\mathrm{CCR}} \cap F_{\mathrm{BCC}}$, $L \leq 1$ and $U \geq 1$. Then, $L(x, y) \in F_{\mathrm{GRS(L,U)}}$ and $U(x, y) \in F_{\mathrm{GRS(L,U)}}$.*

Proof. We can complete the proof in a way similar to Theorem 3.3.2. □

Chapter 4

IMPROVEMENTS FOR INEFFICIENT DMUS

In this chapter, we calculate the improvements for inefficient DMUs in the CCR model. To calculate the improvements, we need to obtain all equations forming F_{CCR}. Moreover, in order to obtain a more flexible improvement, we introduce the equations forming the efficient frontiers of other models. Therefore, before calculating the improvements, we discuss the methods to obtain all equations forming the efficient frontiers. In Section 4.1, we introduce an algorithm for calculating the equations forming F_{CCR} proposed in [24]. In Section 4.2, we propose three kinds of algorithms to calculate all equations forming the efficient frontiers. In Section 4.3, we introduce the previous research formulated by Aparicio, Ruiz and Sirvent [3] as a mixed integer linear programming problem for calculating improvements. In contrast, we propose four types of improvements by utilizing the equations in Section 4.4. In Section 4.5, we show a numerical experiment.

4.1 Algorithm for calculating the equations forming the efficient frontier by solving mixed integer linear programming problems

Analysis of DMUs by calculating the equations forming the efficient frontiers has been considered by Jahanshahloo, Lotfi and Zohrehbandian [24]. They have proposed the following algorithm which is formulated as a mixed integer linear programming problem to obtain the equations forming F_{CCR}.

Algorithm QC

Step 0

Set $J,\ J_1,\ J_2$ as follows:

$$J = \{1, ..., n\},$$

$$J_1 = \{i_1, ..., i_m\},\ i_h := \{e_h, 0\},\ e_h \in \mathbb{R}^m,\ e_{h,j} = 0\ \ (\forall j \neq h),\ e_{h,h} = 1,$$

$$J_2 = \{o_1, ..., o_s\},\ o_l := \{0, e_l\},\ e_l \in \mathbb{R}^s,\ e_{l,j} = 0\ \ (\forall j \neq l),\ e_{l,l} = 1.$$

Set $k := 1$ go to Step 1

Step 1

Solve Problem (Q_k).

$$(\mathrm{Q}(k)) \left\{ \begin{array}{ll} \text{maximize} & \displaystyle\sum_{j \in J \cup J_1 \cup J_2} Q_j \\ \text{subject to} & \displaystyle\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0,\ j \in J \cup J_1, \\ & \displaystyle\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \geq -\mathrm{M}(1 - Q_j),\ j \in J \cup J_1, \\ & \displaystyle\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \geq 0,\ j \in J_2, \\ & \displaystyle\sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq \mathrm{M}(1 - Q_j),\ j \in J_2, \\ & \displaystyle\sum_{r=1}^{s} u_r + \sum_{i=1}^{m} v_i \geq 1, \\ & \displaystyle\sum_{i \notin G_j} Q_i \geq 1,\ j = 1, \ldots, k-1, \\ & \displaystyle\sum_{i \in J \cup J_1} Q_i \geq 1, \\ & Q_j \in \{0, 1\},\ j \in J \cup J_1 \cup J_2, \\ & u_r \geq 0,\ r = 1, \ldots, s, \\ & v_i \geq 0,\ i = 1, \ldots, m, \end{array} \right.$$

where M is a large enough positive number.

If Problem (Q_k) is feasible, set $G_k := \{\text{DMU}(j) : \text{DMU}(j) \text{ lies on the } F_k\}$, where $F_k = \left\{(x, y) : \sum_{r=1}^{s} u_r^* y_r - \sum_{i=1}^{m} v_i^* x_i = 0\right\}$ is a facet of F_{CCR}, $k \leftarrow k + 1$ go to Step 1. If (Q_k) is not feasible, then stop the algorithm.

However, there is no guarantee that all equations are obtained. Therefore, in this thesis, we propose three kinds of algorithms to ensure obtaining all equations forming the efficient frontiers.

4.2 Algorithm for calculating all equations forming the efficient frontiers by utilizing the properties of the polar set

In this section, we propose three kinds of algorithms to ensure obtaining all equations forming the efficient frontiers. The following algorithm is used to construct all equations forming F_{CCR}, F_{BCC}, F_{IRS} and F_{DRS}. By the definitions of the efficient frontiers, $F_{\text{CCR}} \cup F_{\text{BCC}} = F_{\text{IRS}} \cup F_{\text{DRS}}$. Therefore, we construct the algorithm based on properties of the CCR and the BCC models.

Algorithm FFA

Step 0

Set $P(i)\,(i = 1, \ldots, 2n)$ and $P'(i)\,(i = 1, \ldots, 2n + m + s)$ as follows.

$$P(i) := \begin{cases} (x(i)^\top, y(i)^\top)^\top & \text{if } i \in \{1, \ldots, n\}, \\ 2P(i - n) & \text{if } i \in \{n + 1, \ldots, 2n\}. \end{cases} \tag{4.1}$$

$$P'(i) := \begin{cases} P(i) - G & \text{if } i \in \{1, \ldots, 2n\}, \\ e^{i-2n} & \text{if } i \in \{2n + 1, \ldots, 2n + m + s\}, \end{cases} \tag{4.2}$$

where $G := \frac{1}{2n}(P(1) + \cdots + P(2n))$ and e^j is a vector of $\mathbb{R}^{m+s}$ satisfying $e_j^j = 1$ and $e_i^j = 0$ for each $j \in \{1, \ldots, m+s\}$ and $i \in \{1, \ldots, m+s\} \setminus \{j\}$. Let $c_i := i$ for each $i \in \{1, \ldots, m + s\}$ and $\bar{n} := 2n + m + s$. Set $t = 1$ and go to Step 1.

Step 1

If $\dim(\text{co}(\{P'(c_i) : i = 1, \ldots, m+s\})) = m+s$, then go to Step 2. Otherwise, go to Step 3.

Step 2

Step 2–0

Calculate W by solving the following system of linear equations:

$$\begin{cases} (P'(c_1))^\top W = \alpha(c_1), \\ \quad\vdots \\ (P'(c_{m+s}))^\top W = \alpha(c_{m+s}). \end{cases}$$

where $\alpha(c_i)$ $(i = 1, \dots, m+s)$ are as follows.

$$\alpha(c_i) := \begin{cases} 1 & \text{if } c_i \in \{1, \dots, 2n\}, \\ 0 & \text{if } c_i \in \{2n+1, \dots, \bar{n}\}. \end{cases}$$

Step 2–1

If W calculated at Step 2–0 satisfies the following conditions, then $V_t := W$ and $t \leftarrow t+1$.

$$\begin{cases} (P'(j))^\top W \leq 1, \; j = 1, \dots, 2n, \\ W_i \leq 0, \; i = 1, \dots, m, \\ W_i \geq 0, \; i = m+1, \dots, m+s. \end{cases}$$

Otherwise, $\{V_1, \dots, V_t\}$ remain. If $c_1 = 2n-m-s+1$, go to Step 4. Otherwise, go to Step 3.

Step 3

Step 3–0

Set $c_{m+s} \leftarrow c_{m+s} + 1$ and $j := m+s$. Go to Step 3–1.

Step 3–1

If $c_j \leq 2n-m-s+j$, set $c_{j'} \leftarrow c_j + j' - j$ for every $j' > j$. Go to Step 1. Otherwise, set $c_{j-1} \leftarrow c_{j-1} + 1, j \leftarrow j-1$ and go to Step 3–1.

Step 4

For each $i \in \{1, \dots, t-1\}$, let $(-p_i^\top, q_i^\top)^\top := V_i$, where $p_i \in \mathbb{R}^m$ and $q_i \in \mathbb{R}^s$. For each $i = 1, \dots, t-1$, if $\max\{q_{i,1}, \dots, q_{i,s}\} > 0$ and $\frac{1+(-p_i^\top, q_i^\top)^\top G}{\max\{q_{i,1}, \dots, q_{i,s}\}} > 0$, then $c_i := \frac{1+(-p_i^\top, q_i^\top)^\top G}{2}$. Otherwise, $c_i := 1 + (-p_i^\top, q_i^\top)^\top G$. Then, the hyperplane forming the efficient frontier is as follows.

$$H_{p_i, q_i, c_i} := \{(x, y) : -p_i^\top x + q_i^\top y = c_i\}.$$

Stop the algorithm.

At Step 0, in order to obtain all equations forming F_{CCR}, for each $i = 1, \ldots, n$, $P(i+n)$ is generated. Let $\bar{P} := \mathrm{co}(\{P'(1), \ldots, P'(2n)\})$. To calculate all vertices of $(\bar{P})^* \cap \hat{\mathbb{R}}^{m+s} := \{Z \in \mathbb{R}^{m+s} : Z_i \leq 0\,(1 \leq i \leq m),\ Z_i \geq 0\,(m+1 \leq i \leq m+s)\}$, all combinations of $\{P'(1), \ldots, P'(\bar{n})\}$ are considered. At Step 1, to examine whether there exists a solution of the linear system at Step 2–0, $\dim\{P'(c_i) : i = 1, \ldots, m{+}s\}$ is calculated. At Step 2, to examine whether W obtained at Step 2–0 is a vertex of $(\bar{P})^*$, $W^\top P'(1), \ldots, W^\top P'(\bar{n})$ are calculated. If all values of $W^\top P'(1), \ldots, W^\top P'(\bar{n})$ are less than or equal to one, then W is a vertex of $(\bar{P})^*$. At Step 3, to select all combinations of choosing $m+s$ numbers from $\{1, \ldots, \bar{n}\}$, $c_1, \ldots, c_{m+s}$ are updated. At Step 4, for each $i \in \{1, \ldots, t-1\}$, the necessity of H_{p_i, q_i, c_i} for constructing the efficient frontier is examined.

Example 4.2.1. We illustrate Algorithm FFA in the case of $m = s = 1$. The data of DMUs is listed in Table 4.1 and illustrated in Figure 4.1. By executing Algorithm FFA, we can construct all equations forming F_{CCR}, F_{BCC}, F_{IRS} and F_{DRS} based on the data in Table 4.1.

Table 4.1: The data of four DMUs

DMU	A	B	C	D
Input	2	4	4	6
Output	1	2	3	2

Step 0: Since $n = 4$, according to (4.1), $P_i\,(i = 1, \ldots, 8)$ shown in Figure 4.2 are calculated as follows:

$$P_1 = (2,1)^\top, P_2 = (4,2)^\top, P_3 = (4,3)^\top, P_4 = (6,2)^\top,$$

$$P_5 = 2P_1 = (4,2)^\top, P_6 = (8,4)^\top, P_7 = (8,6)^\top, P_8 = (12,4)^\top.$$

Then, $G = \frac{1}{2n}(P_1 + \cdots + P_{2n}) = \frac{1}{8}(48, 24)^\top = (6,3)^\top$. According to (4.2), $P_i'\,(i = 1, \ldots, 10)$ shown in Figure 4.3 are calculated as follows:

$$P_1' = (-4,-2)^\top, P_2' = (-2,-1)^\top, P_3' = (-2,0)^\top, P_4' = (0,-1)^\top, P_5' = (-2,-1)^\top,$$

$$P_6' = (2,1)^\top, P_7' = (2,3)^\top, P_8' = (6,1)^\top, P_9' = (1,0)^\top, P_{10}' = (0,1)^\top.$$

Set $c_1 := 1, c_2 := 2$, $\bar{n} := 2n + m + s = 10$ and $t := 1$. Go to Step 1.

Step 1: Since $\dim\{(-4,-2)^\top, (-2,-1)^\top\} = 1$, go to Step 3.

Step 3: Set $c_1 := 1, c_2 := 3$. Go to Step 1.

Step 1: Since $\dim\{(-4,-2)^\top, (-2,0)^\top\} = 2$, go to Step 2.

Step 2: Calculate W satisfying the following linear system:

$$\begin{cases} (-4,-2)^\top W = 1, \\ (-2,0)^\top W = 1. \end{cases}$$

Then, we obtain $W = (-\frac{1}{2}, \frac{1}{2})^\top$. We examine whether W is a vertex of $\bar{P} := \mathrm{co}(\{P_1', \ldots, P_8'\})$. Since, $(P_j')^\top(-\frac{1}{2}, \frac{1}{2}) \leq 1\,(j = 1, \ldots, 8)$, $(-\frac{1}{2}, \frac{1}{2})^\top$ is a vertex of $(\bar{P})^*$. By Lemma 2.1.5 and the coordinate transformation moving G to the origin, we can obtain all equations forming $P := \mathrm{co}(\{P_1, \ldots, P_8\})$. In order to obtain only the efficient facets of P, we consider the vertices contained in $\{W \in \mathbb{R}^2 : W_1 \leq 0,\, W_2 \geq 0\}$. Since, $W_1 = -\frac{1}{2} \leq 0$ and $W_2 = \frac{1}{2} \geq 0$, set $V_1 := (-\frac{1}{2}, \frac{1}{2})^\top$ which is a vertex of polytope Q shown in Figure 4.4. Set $t := 2$. Since $c_1 \neq 2n - m - s + 1 = 7$, go to Step 3.

Step 3: Set $c_1 := 1, c_2 := 4$. Go to Step 1.

We repeat this operation to $c_1 = 9, c_2 = 10$. Then, $t = 4$. $V_1 = (-\frac{1}{2}, \frac{1}{2})^\top$, $V_2 = (-\frac{1}{4}, 0)^\top$, $V_3 = (-\frac{1}{2}, \frac{2}{3})^\top$ and $V_4 = (0, \frac{1}{3})^\top$ are all vertices except the origin of Q. Go to Step 4.

Step 4: For $t = 1$, $-\frac{1}{2}x + \frac{1}{2}y = 1 + (-\frac{1}{2}, \frac{1}{2})^\top(6,3) = -\frac{1}{2}$. Hence, $H_1 := \{(x,y) : -x + y = -1\}$. Similarly, $H_2 := \{(x,y) : x = 2\}$ and $H_3 := \{(x,y) : -\frac{1}{2}x + \frac{2}{3}y = 0\}$. For $t = 4$, since $|Y_{4,1}| = \frac{1}{3} > 0$ and $\frac{1+(0,\frac{1}{3})(6,3)}{Y_{4,1}} = 6 > 0$, $\frac{1}{3}y = \frac{1+(0,\frac{1}{3})^\top(6,3)}{2} = 1$. Hence, $H_4 := \{(x,y) : y = 3\}$. $H_1, \ldots, H_4$ are all efficient facets of P. Then stop the algorithm.

By Algorithm FFA, we can obtain four vertices of Q shown in Figure 4.4 as follows:

$$V_1 = \left(-\frac{1}{2}, \frac{1}{2}\right)^\top, V_2 = \left(-\frac{1}{4}, 0\right)^\top, V_3 = \left(-\frac{1}{2}, \frac{2}{3}\right)^\top, V_4 = \left(0, \frac{1}{3}\right)^\top.$$

By Lemma 2.1.5, Q^* shown in Figure 4.5 are formed by four equations as follows:

$$H_1' : \left(-\frac{1}{2}, \frac{1}{2}\right)^\top (x,y) = 1, H_2' : \left(-\frac{1}{4}, 0\right)^\top (x,y) = 1,$$

$$H_3' : \left(-\frac{1}{2}, \frac{2}{3}\right)^\top (x,y) = 1, H_4' : \left(0, \frac{1}{3}\right)^\top (x,y) = 1.$$

By the coordinate transformation, we obtain four equations depicted in Figure 4.6 as follows:

$$H_1'' : \left(-\frac{1}{2},\frac{1}{2}\right)^\top (x,y) = 1 + \left(-\frac{1}{2},\frac{1}{2}\right)^\top (6,3) = -\frac{1}{2},$$
$$H_2'' : \left(-\frac{1}{4},0\right)^\top (x,y) = 1 + \left(-\frac{1}{4},0\right)^\top (6,3) = -\frac{1}{2},$$
$$H_3'' : \left(-\frac{1}{2},\frac{2}{3}\right)^\top (x,y) = 1 + \left(-\frac{1}{2},\frac{2}{3}\right)^\top (6,3) = 0,$$
$$H_4'' : \left(0,\frac{1}{3}\right)^\top (x,y) = 1 + \left(0,\frac{1}{3}\right)^\top (6,3) = 2.$$

By the operation at Step 5, we obtain four equations illustrated in Figure 4.7 as follows:

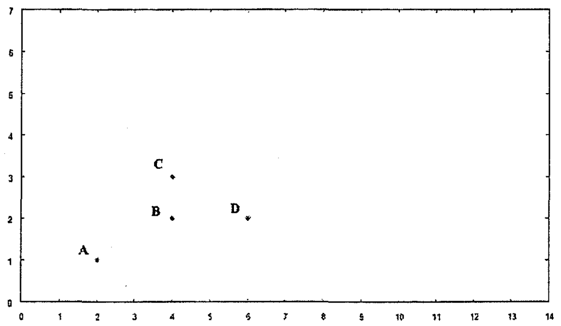
$$H_1 : \left(-\frac{1}{2},\frac{1}{2}\right)^\top (x,y) = -\frac{1}{2}, H_2 : \left(-\frac{1}{4},0\right)^\top (x,y) = -\frac{1}{2},$$
$$H_3 : \left(-\frac{1}{2},\frac{2}{3}\right)^\top (x,y) = 0, H_4 : \left(0,\frac{1}{3}\right)^\top (x,y) = \frac{1+(0,\frac{1}{3})^\top(6,3)}{2} = 1.$$

Then, $H_1, \ldots, H_4$ form the efficient frontiers.

Since the T_{CCR} is a closed convex cone, by the operation at Step 0, we can always calculate all equations of the CCR model. Moreover, the origin is contained in $\bar{P}$. Figure 4.4 shows the hyperplane $\{(x,y) : (P'(j))^\top(x,y) = 1\}$ for each $j = 1, \ldots, 2n$. Polytope Q is the intersection of $\hat{\mathbb{R}}^{m+s}$ and $(\bar{P})^*$. We calculate all vertices of Q by performing from Step 1 to Step 3. Figure 4.5 shows the hyperplane $\{(x,y) : v^\top(x,y) = 1\}$ for each vertex v of polytope Q except the origin shown in Figure 4.4. By the coordinate transformation moving G to the origin, we get Figure 4.6. Figure 4.7 shows all the hyperplanes calculated by Algorithm FFA. By the operation at Step 4, the hyperplane consisting of only DMUs generated at Step 0 is replaced by the hyperplane consisting of original DMUs. For example, the hyperplane consisting of DMU$(n+2)$ and DMU$(n+4)$ is replaced by the hyperplane consisting of DMU(2) and DMU(4).

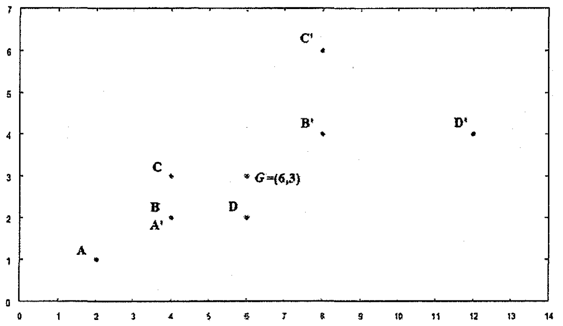
Theorem 4.2.1. *The intersection of $\bar{P}^*$ and $\hat{\mathbb{R}}^{m+s}$ is a polytope containing* 0.

Proof. By the definition of $\bar{P}$ and Assumption (A4), $\bar{P}$ is a polytope and $0 \in \operatorname{int} \bar{P}$. Hence, $(\bar{P})^*$ is a polytope and $0 \in \operatorname{int}(\bar{P})^*$. Of course, $\hat{\mathbb{R}}^{m+s}$ is a closed convex polyhedral set containing 0. Thus, the intersection of $\hat{\mathbb{R}}^{m+s}$ and $(\bar{P})^*$ is a polytope containing 0. □



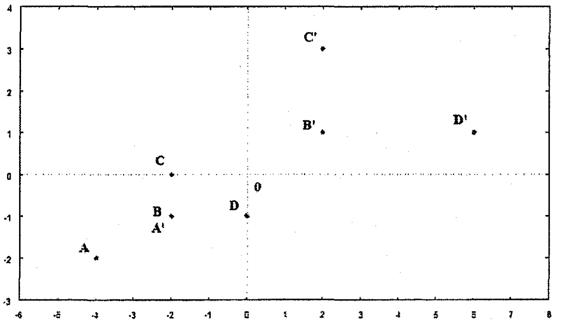
$P_1 := \mathrm{A}$
$P_2 := \mathrm{B}$
$P_3 := \mathrm{C}$
$P_4 := \mathrm{D}$

Figure 4.1: Illustration of all DMUs



$P_5 := 2P_1 = \mathrm{A}'$
$P_6 := 2P_2 = \mathrm{B}'$
$P_7 := 2P_3 = \mathrm{C}'$
$P_8 := 2P_4 = \mathrm{D}'$

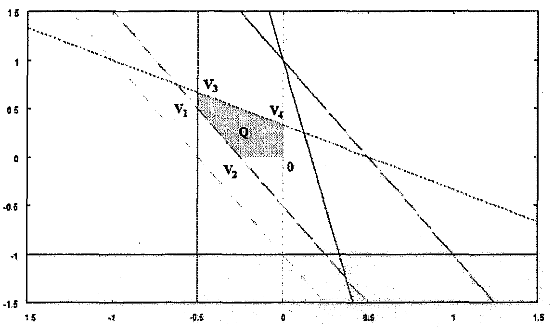
Figure 4.2: We add two times the original DMUs



Generation of $P_1', \ldots, P_8'$.
$P_1' := P_1 - G$
$P_2' := P_2 - G$
$P_3' := P_3 - G$
$P_4' := P_4 - G$
$P_5' := P_5 - G$
$P_6' := P_6 - G$
$P_7' := P_7 - G$
$P_8' := P_8 - G$

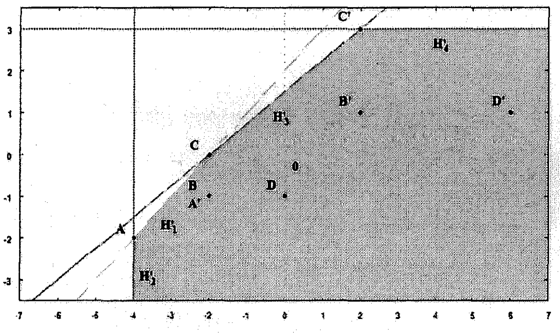
Figure 4.3: The coordinate transformation moving G to the origin

By Theorem 4.2.1, the number of vertices of the intersection of $\bar{P}^*$ and $\hat{\mathbb{R}}^{m+s}$ is finite. In particular, at Step 3, all combinations of $c_1, \ldots, c_{m+s}$ from $\{1, \ldots, \bar{n}\}$ are selected. Thus, Algorithm FFA terminates within ${}_{\bar{n}}C_{m+s}$ iterations. Let h be the number of hyperplanes H_{p_i,q_i,c_i} calculated by Algorithm FFA. For each $j = 1, \ldots, h$,



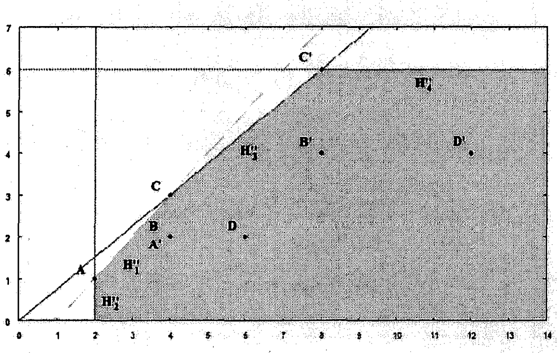
$$
\begin{aligned}
V_1 &:= (-\tfrac{1}{2}, \tfrac{1}{2}) \\
V_2 &:= (-\tfrac{1}{4}, 0) \\
V_3 &:= (-\tfrac{1}{2}, \tfrac{2}{3}) \\
V_4 &:= (0, \tfrac{1}{3})
\end{aligned}
$$

Figure 4.4: Hyperplane that inner product of each DMU and (x, y) equals one



$$
\begin{aligned}
H_1' &: (-\tfrac{1}{2}, \tfrac{1}{2})^\top (x, y) = 1 \\
H_2' &: (-\tfrac{1}{4}, 0)^\top (x, y) = 1 \\
H_3' &: (-\tfrac{1}{2}, \tfrac{2}{3})^\top (x, y) = 1 \\
H_4' &: (0, \tfrac{1}{3})^\top (x, y) = 1
\end{aligned}
$$

Figure 4.5: The polar set of Q



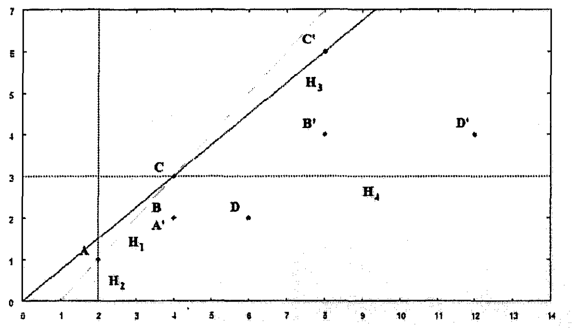
$$
\begin{aligned}
H_1'' &: (-\tfrac{1}{2}, \tfrac{1}{2})^\top (x, y) = -\tfrac{1}{2} \\
H_2'' &: (-\tfrac{1}{4}, 0)^\top (x, y) = -\tfrac{1}{2} \\
H_3'' &: (-\tfrac{1}{2}, \tfrac{2}{3})^\top (x, y) = 0 \\
H_4'' &: (0, \tfrac{1}{3})^\top (x, y) = 2
\end{aligned}
$$

Figure 4.6: The coordinate transformation

let

$$
\begin{aligned}
W_j &:= (-p_j^\top, q_j^\top)^\top, && (4.3) \\
S_c &:= \{i \in \{1, \ldots, h\} : H_{p_i, q_i, c_i} \cap T_{\text{CCR}} \subset F_{\text{CCR}}\}, && (4.4) \\
S_b &:= \{i \in \{1, \ldots, h\} : H_{p_i, q_i, c_i} \cap T_{\text{BCC}} \subset F_{\text{BCC}}\}. && (4.5)
\end{aligned}
$$

Then, T_{CCR} and F_{CCR} can be represented by using coefficients of equations as follows.



$$H_1 : (-\tfrac{1}{2}, \tfrac{1}{2})^\top (x, y) = -\tfrac{1}{2}$$
$$H_2 : (-\tfrac{1}{4}, 0)^\top (x, y) = -\tfrac{1}{2}$$
$$H_3 : (-\tfrac{1}{2}, \tfrac{2}{3})^\top (x, y) = 0$$
$$H_4 : (0, \tfrac{1}{3})^\top (x, y) = 1$$

Figure 4.7: All hyperplanes obtained by Algorithm FFA

Theorem 4.2.2. $T_{\mathrm{CCR}} = \bigcap\limits_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$.

Proof. First, we shall show that $T_{\mathrm{CCR}} \subset \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$. For each $Z := (x^\top, y^\top)^\top \in T_{\mathrm{CCR}}$, there exists $\lambda' \geq 0$ such that $x \geq \sum_{i=1}^n \lambda_i' x(i)$, $y \leq \sum_{i=1}^n \lambda_i' y(i)$. Since $W_j = (-p_j^\top, q_j^\top)^\top$ $(p_j \geq 0,\ q_j \geq 0)$, $W_j^\top Z = -p_j^\top x + q_j^\top y \leq -p_j^\top \sum_{i=1}^n \lambda_i' x(i) + q_j^\top \sum_{i=1}^n \lambda_i' y(i)$. By the definition of F_{CCR}, $-p_j^\top x(i) + q_j^\top y(i) \leq 0$ for each $i \in \{1, \ldots, n\}$. Hence, $W_j^\top Z \leq 0$ and $(x^\top, y^\top)^\top \in \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$. Therefore, $T_{\mathrm{CCR}} \subset \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$.

Second, we shall show that $T_{\mathrm{CCR}} \supset \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$. For each $Z \in \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$, the following two cases occur.

(i) There exists $j \in S_c$ such that $W_j^\top Z = 0$.

(ii) There exist no $j \in S_c$ such that $W_j^\top Z = 0$.

In Case (i), by the definition of W_j, there exists $\lambda \geq 0$ such that $x = \sum_{i=1}^n \lambda_i x(i)$, $y = \sum_{i=1}^n \lambda_i y(i)$. Hence, $Z \in T_{\mathrm{CCR}}$. In Case (ii), there exist $\delta > 0$ and $j \in S_c$ such that $W_j^\top (Z + \delta W_j) = 0$ and $W_k^\top (Z + \delta W_k) \leq 0$ for each $k \in S_c$. Let $Z' := Z + \delta W_j$. Then, $x \geq x'$ and $y \leq y'$. By the definition of W_j, there exists $\lambda \geq 0$ such that $x' = \sum_{i=1}^n \lambda_i x(i)$, $y' = \sum_{i=1}^n \lambda_i y(i)$. Hence, $Z' \in T_{\mathrm{CCR}}$ and $Z \in T_{\mathrm{CCR}}$. Therefore, $T_{\mathrm{CCR}} \supset \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$. Consequently, $T_{\mathrm{CCR}} = \bigcap_{j \in S_c} \{Z : W_j^\top Z \leq 0\}$. □

Theorem 4.2.3. $T_{\mathrm{BCC}} = \bigcap\limits_{j \in S_b} \{Z : W_j^\top Z \leq c_j\}$.

Proof. We can complete the proof in a way similar to Theorem 4.2.2. □

Theorem 4.2.4. $F_{\mathrm{CCR}} = \Big(\bigcup\limits_{j \in S_c} \{Z : W_j^\top Z = 0\} \Big) \cap T_{\mathrm{CCR}}$.

Proof. First, we shall show that $F_{\text{CCR}} \subset \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$. For each $Z' := (x'^\top, y'^\top)^\top \in F_{\text{CCR}}$, $(x'^\top, y'^\top)^\top \in T_{\text{CCR}}$. Let $(\theta^*_{\text{CCR}}(Z'), \lambda^*_1, \ldots, \lambda^*_n)$ be an optimal solution of the CCR model for Z', that is $\theta^*_{\text{CCR}}(Z')$ solves the following problem.

$$
(\text{CCRD}(\text{Z}'))\begin{cases}
\text{minimize} & \theta \\
\text{subject to} & \theta x'_i - \displaystyle\sum_{j=1}^{n}\lambda_j x(j)_i \geq 0,\ i = 1, \ldots, m, \\
& \displaystyle\sum_{j=1}^{n}\lambda_j y(j)_r - y'_r \geq 0,\ r = 1, \ldots, s, \\
& \lambda_j \geq 0,\ j = 1, \ldots, n, \\
& \theta \in \mathbb{R}.
\end{cases}
$$

Since $\theta^*_{\text{CCR}}(Z') = 1$, there exists i such that $x'_i = \sum_{j=1}^{n}\lambda^*_j x(j)_i$. Hence, $(x'^\top, y'^\top)^\top \in \text{bd}(T_{\text{CCR}})$. By Theorem 4.2.2, there exists $j \in S_c$ such that $W_j^\top Z' = 0$. Hence, $Z' \in \bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}$. Therefore, $F_{\text{CCR}} \subset \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$.

Second, we shall show that $F_{\text{CCR}} \supset \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$. For each $Z' \in \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$, by Theorem 4.2.2, $Z' \in \text{bd}(T_{\text{CCR}})$. By definition of F_{CCR}, $Z' \in F_{\text{CCR}}$. Therefore, $F_{\text{CCR}} \supset \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$. Consequently, $F_{\text{CCR}} = \left(\bigcup_{j\in S_c}\{Z : W_j^\top Z = 0\}\right) \cap T_{\text{CCR}}$. □

Theorem 4.2.5. $F_{\text{BCC}} = \left(\displaystyle\bigcup_{j\in S_b}\{Z : W_j^\top Z = c_j\}\right) \cap T_{\text{BCC}}$.

Proof. We can complete the proof in a way similar to Theorem 4.2.4. □

By Algorithm FFA, we obtain all equations forming the efficient frontiers of the four models. We classify the equations under the following theorems.

Theorem 4.2.6. *Assume that $H_{p,q,c} = \{(x, y) \in \mathbb{R}^{m+s} : q^\top y - p^\top x = c\}$ is calculated by Algorithm* FFA. *If $c = 0$, then $H_{p,q,c} \cap T_{\text{CCR}}$ is a facet of T_{CCR}.*

Proof. Since p, q and c are constructed at Step 2 of Algorithm FFA, $\dim H_{p,q,c} = m+s-1$. By Assumption (A4), $\dim T_{\text{CCR}} = m+s$. By Lemma 2.1.6, $\dim H_{p,q,c} \cap (\bar{P} + G) = m + s - 1$. By the definition of T_{CCR}, $\bar{P} + G \subset T_{\text{CCR}}$. Therefore, $\dim H_{p,q,c} \cap T_{\text{CCR}} = m + s - 1$. At Step 2 of Algorithm FFA, $(-p^\top, q^\top)^\top (x(j)^\top, y(j)^\top) \leq 0$ $(j = 1, \ldots, n)$. For each $(x^\top, y^\top)^\top \in T_{\text{CCR}}$, there exists $\lambda' \geq 0$ such that $x \geq$

$\sum_{j=1}^{n} \lambda'_j x(j)$, $0 \leq y \leq \sum_{j=1}^{n} \lambda'_j y(j)$. Then $(-p^\top, q^\top)(x^\top, y^\top)^\top = -p^\top x + q^\top y \leq -p^\top \sum_{j=1}^{n} \lambda'_j x(j) + q^\top \sum_{j=1}^{n} \lambda'_j y(j) \leq 0$. Consequently, $H_{p,q,c} \cap T_{\mathrm{CCR}}$ is a facet of T_{CCR}. □

Theorem 4.2.7. *If $c \neq 0$, then $H_{p,q,c} \cap T_{\mathrm{BCC}}$ is a facet of T_{BCC}.*

Proof. By Assumption (A4), $\dim T_{\mathrm{BCC}} = m+s$. By Lemma 2.1.6 and the operation at Step 4 of Algorithm FFA, $\dim H_{p,q,c} \cap \mathrm{co}(\{P_1, \ldots, P_n\}) = m+s-1$. By the definition of T_{BCC}, $\mathrm{co}(\{P_1, \ldots, P_n\}) \subset T_{\mathrm{BCC}}$. Therefore, $\dim H_{p,q,c} \cap T_{\mathrm{BCC}} = m+s-1$. At Step 2 of Algorithm FFA, $(-p^\top, q^\top)^\top (x(j)^\top, y(j)^\top) \leq c$ $(j = 1, \ldots, n)$. For each $(x^\top, y^\top)^\top \in T_{\mathrm{BCC}}$, there exists $\lambda' \geq 0$ such that $x \geq \sum_{j=1}^{n} \lambda'_j x(j)$, $0 \leq y \leq \sum_{j=1}^{n} \lambda'_j y(j)$, $\sum_{j=1}^{n} \lambda'_j = 1$. Then $(-p^\top, q^\top)(x^\top, y^\top)^\top = -p^\top x + q^\top y \leq -p^\top \sum_{j=1}^{n} \lambda'_j x(j) + q^\top \sum_{j=1}^{n} \lambda'_j y(j) \leq c$. Consequently, $H_{p,q,c} \cap T_{\mathrm{BCC}}$ is a facet of T_{BCC}. □

Theorem 4.2.8. *If $c \leq 0$, then $H_{p,q,c} \cap T_{\mathrm{IRS}}$ is a facet of T_{IRS}.*

Proof. We can complete the proof in a way similar to Theorem 4.2.7. □

Theorem 4.2.9. *If $c \geq 0$, then $H_{p,q,c} \cap T_{\mathrm{DRS}}$ is a facet of T_{DRS}.*

Proof. We can complete the proof in a way similar to Theorem 4.2.7. □

Theorem 4.2.10. *If $c = 0$ and $\dim(\{(x(i)^\top, y(i)^\top)^\top : i = 1, \ldots, n\} \cap H_{p,q,c}) = m+s-1$, then $H_{p,q,c} \cap T_{\mathrm{BCC}}$ is a facet of T_{BCC}.*

Proof. Since $\dim(\{(x(i)^\top, y(i)^\top)^\top : i = 1, \ldots, n\} \cap H_{p,q,c}) = m+s-1$, $\dim H_{p,q,c} \cap T_{\mathrm{BCC}} = m+s-1$. At Step 2 of Algorithm FFA, $(-p^\top, q^\top)^\top (x(j)^\top, y(j)^\top) \leq 0$ $(j = 1, \ldots, n)$. For each $(x^\top, y^\top)^\top \in T_{\mathrm{BCC}}$, there exists $\lambda' \geq 0$ such that $x \geq \sum_{j=1}^{n} \lambda'_j x(j)$, $0 \leq y \leq \sum_{j=1}^{n} \lambda'_j y(j)$, $\sum_{j=1}^{n} \lambda'_j = 1$. Then $(-p^\top, q^\top)(x^\top, y^\top)^\top = -p^\top x + q^\top y \leq -p^\top \sum_{j=1}^{n} \lambda'_j x(j) + q^\top \sum_{j=1}^{n} \lambda'_j y(j) \leq 0$. Therefore, $H_{p,q,c} \cap T_{\mathrm{BCC}}$ is a facet of T_{BCC}. □

By classifying the equations in accord with the above theorems, we can obtain efficiency scores of the four models easily by substituting the input and output values of each DMU as follows. Let S_c, S_b, S_i and S_d be the index sets of all hyperplanes of the CCR, BCC, IRS and DRS models, respectively.

Theorem 4.2.11. (Jahanshahloo, Lotfi and Zohrehbandian [24]) *Let H_{p_j,q_j,c_j} be a hyperplane forming the efficient frontier of the* CCR *model for each $j \in S_c$, where $H_{p_j,q_j,c_j} := \{(x,y) : -p_j^\top x + q_j^\top y = c_j\}$, then the efficiency score of* DMU(k) *in the* CCR *model is obtained as follows.*

$$\text{Eff}(\text{DMU}(k)) = \max\left\{\frac{q_j^\top y(k)}{p_j^\top x(k)} : j \in S_c\right\}.$$

Theorem 4.2.12. (Jahanshahloo, Lotfi and Zohrehbandian [24]) *Let H_{p_j,q_j,c_j} be a hyperplane forming the efficient frontiers of the* BCC, DRS, IRS *models for each $j \in S_b,\ S_i,\ S_d$, where $H_{p_j,q_j,c_j} := \{(x,y) : -p_j^\top x + q_j^\top y = c_j\}$, then the efficiency scores of* DMU(k) *in the* BCC, DRS, IRS *models are obtained as follows.*

$$\text{Eff}(\text{DMU}(k)) = \max\left\{\frac{q_j^\top y(k) + c_j}{p_j^\top x(k)} : p_j^\top x(k) \neq 0,\, j \in S_b,\ S_i,\ S_d\right\}.$$

By Jahanshahloo, Lotfi and Zohrehbandian [24], the RTS are obtained as follows. Let h be the number of the hyperplanes calculated by Algorithm FFA. For each $j = 1, ..., h$, H_{p_j,q_j,c_j} be the hyperplane defined by $\{(x,y) : -p_j^\top x + q_j^\top y = c_j\}$. Let $\theta^*_{\text{BCC}}(k)$ be the optimal value of Problem (BCCLP(k)) for DMU(k) and

$$S(k) := \left\{c_j : \frac{q_j^\top y(k) + c_j}{p_j^\top x(k)} = \theta^*_{\text{BCC}(k)},\, p_j^\top x(k) \neq 0,\, j \in S_b\right\}.$$

Theorem 4.2.13. (Jahanshahloo, Lotfi and Zohrehbandian [24]) *The* RTS *is classified as the following.*

(i) DMU(k) is said to be the increasing RTS if $\min\{S(k)\} < \max\{S(k)\} \leq 0$ or $\min\{S(k)\} = \max\{S(k)\} < 0$.

(ii) DMU(k) is said to be the decreasing RTS if $0 \leq \min\{S(k)\} < \max\{S(k)\}$ or $0 < \min\{S(k)\} = \max\{S(k)\}$.

(iii) DMU(k) is said to be the constant RTS if $\min\{S(k)\} < 0 < \max\{S(k)\}$ or $\min\{S(k)\} = \max\{S(k)\} = 0$.

For the data in Table 4.1, we identify the RTS under Theorem 4.2.13. Table 4.2 shows the classification of the RTS. In general, the RTS is considered for only BCC-efficient DMUs, but by using Theorem 4.2.13, we can also consider the RTS for BCC-inefficient DMUs based on the points projected on F_{BCC}.

Table 4.2: Classification of the RTS

DMU	$\min\{S(k)\}$	$\max\{S(k)\}$	RTS
A	-2	-1	increasing
B	-1	-1	increasing
C	-1	3	constant
D	-1	-1	increasing

As the numbers of inputs, outputs and DMUs increase, Algorithm FFA requires a large number of iterations to construct all equations forming the efficient frontiers. For example, in the case where $m = 2$, $s = 1$ and $n = 10$, the number of iterations of Algorithm FFA is 1,771. In the case where $m = 3$, $s = 2$ and $n = 20$, the number of iterations of Algorithm FFA is 1,221,759. Therefore, we devise a method for reducing the number of iterations. If $x(a)_i < x(b)_i\,(i = 1, \ldots, m)$ and $y(a)_i > y(b)_i\,(i = 1, \ldots, s)$, then DMU$(b)$ can be removed from the original DMUs. In Table 4.3, the averages and dispersions of the computational times of Algorithm FFA and modified Algorithm FFA for 20 test problems on $n = 10, 20, \ldots, 80$ are listed. The input and output values are randomly-determined. By modifying Algorithm FFA, we can calculate all equations forming the efficient frontier in a realistic time.

Table 4.3: The averages and dispersions of the computational times (seconds)

	Algorithm FFA		modified Algorithm FFA	
number of DMUs	averages	dispersions	averages	dispersions
10	0.531	0.001	0.240	0.012
20	11.949	0.046	1.815	1.505
30	92.197	0.911	9.994	86.556
40	393.139	16.214	31.924	821.405
50	1338.005	53.527	51.289	2479.179
60	3615.760	638.221	83.638	4106.046
70	7492.604	2060.928	166.074	22895.020
80	17127.256	97780.144	284.729	88195.173

(The numbers of input and output are 3 and 2, respectively.)

Next, we propose an algorithm to obtain all equations forming the efficient frontier of only the CCR model. When we calculate cross efficiency scores in Chapter 5, we need to use only the equations forming F_{CCR}. Therefore, we propose another algorithm for constructing all equations forming F_{CCR} whose calculation time is smaller than Algorithm FFA. Let $P := \{0, \mathrm{DMU}(1), \ldots, \mathrm{DMU}(n)\}$. Then, $\mathrm{co}(P)$ is contained in T_{CCR} and the intersection of boundary of $\mathrm{co}(P)$ and T_{CCR} is nonempty. Hence, by calculating the equations forming $\mathrm{co}(P)$, we can obtain all equations forming F_{CCR} as follows:

Algorithm FFC

Step 0

Set $P(i)$ $(i = 1, \ldots, n+m+s+1)$ as follows:

$$P(i) := \begin{cases} (x(i)^\top, y(i)^\top)^\top - G & \text{if } i \in \{1, \ldots, n\}, \\ (0, \ldots, 0)^\top - G & \text{if } i = n+1, \\ e^{i-n-1} & \text{if } i \in \{n+2, \ldots, n+m+s+1\}, \end{cases} \tag{4.6}$$

where $G := \frac{1}{n+1}((x(1)^\top, y(1)^\top)^\top + \cdots + (x(n)^\top, y(n)^\top)^\top)$ and e^j is a vector of $\mathbb{R}^{m+s}$ satisfying $e^j_j = 1$ and $e^j_i = 0$ for each $j \in \{1, \ldots, m+s\}$ and $i \in \{1, \ldots, m+s\} \setminus \{j\}$. Let $c_i := i$ for each $i \in \{1, \ldots, m+s\}$ and $t := 1$. Go to Step 1.

Step 1

If $\dim \{P(c_i) : i = 1, \ldots, m+s\} = m+s$, then go to Step 2. Otherwise, go to Step 4.

Step 2

Step 2–0

Calculate W by solving the following system of linear equations:

$$\begin{cases} (P(c_1))^\top W = \alpha(c_1), \\ \quad\vdots \\ (P(c_{m+s}))^\top W = \alpha(c_{m+s}), \end{cases}$$

where,

$$\alpha(c_i) := \begin{cases} 1 & \text{if } c_i \in \{1, \ldots, n+1\}, \\ 0 & \text{if } c_i \in \{n+2, \ldots, n+m+s+1\}. \end{cases}$$

Go to Step 3.

Step 2–1

If W calculated at Step 2–0 satisfies the following conditions, then set $V_t := W$ and $t \leftarrow t+1$.

$$\begin{cases} (P(j))^\top W \leq 1, \; j = 1, \ldots, n+1, \\ W_i \leq 0, \; i = 1, \ldots, m, \\ W_i \geq 0, \; i = m+1, \ldots, m+s, \\ W_{m+1} + \cdots + W_{m+s} > 0. \end{cases}$$

Otherwise, $\{V_1, \ldots, V_t\}$ remain. If $c_1 = n-m-s+2$, go to Step 4. Otherwise, go to Step 3.

Step 3

Step 3–0

Set $c_{m+s} \leftarrow c_{m+s} + 1$ and $j := m+s$. Go to Step 3–1.

Step 3–1

If $c_j \leq n - m - s + 1 + j$, set $c_{j'} \leftarrow c_j + j' - j$ for every $j' > j$ and go to Step 1. Otherwise, set $c_{j-1} \leftarrow c_{j-1} + 1, j \leftarrow j - 1$ and go to Step 3–1.

Step 4

For each $i \in \{1, \ldots, t-1\}$, let $(-p_i^\top, q_i^\top)^\top := V_i$, where $p_i \in \mathbb{R}^m$ and $q_i \in \mathbb{R}^s$. For each $i = 1, \ldots, t-1$, let $c_i := 1 + (-p_i^\top, q_i^\top)^\top G$ and the hyperplane forming the efficient frontier is as follows.

$$H_{p_i, q_i, c_i} := \{(x, y) : -p_i^\top x + q_i^\top y = c_i\}.$$

Stop the algorithm.

The equations calculated by Algorithm FFC are classified by following theorems.

Theorem 4.2.14. *Assume that $H_{p,q,c} = \{(x,y) \in \mathbb{R}^{m+s} : p^\top x + q^\top y = c\}$ is calculated by Algorithm* FFC. *If $c = 0$, then $H_{p,q,c} \cap T_{\text{CCR}}$ is a facet of T_{CCR}.*

Proof. Since p and q are constructed at Step 2 of Algorithm FFC, $p \leq 0, q \geq 0$ and $\dim H_{p,q,c} = m + s - 1$. By Assumption (A4), $\dim T_{\text{CCR}} = m + s$. Let $\bar{P} := \{P(1), \ldots, P(n+1)\}$. By Lemma 2.1.6 and the conditions defined at Step 3 of Algorithm FFC, $\dim H_{p,q,c} \cap (\bar{P} + G) = m+s-1$. By the definition of T_{CCR}, $\bar{P} + G \subset T_{\text{CCR}}$.

Therefore, $\dim H_{p,q,c} \cap T_{\text{CCR}} = m + s - 1$. Let $c = 0$. Then, at Step 3 of Algorithm FFC, $(p^\top, q^\top)^\top (x(j)^\top, y(j)^\top) \leq 0$ $(j = 1, \ldots, n)$. For each $(x^\top, y^\top)^\top \in T_{\text{CCR}}$, there exists $\lambda' \geq 0$ such that $x \geq \sum_{j=1}^{n} \lambda'_j x(j)$, $0 \leq y \leq \sum_{j=1}^{n} \lambda'_j y(j)$. Then $(p^\top, q^\top)^\top (x^\top, y^\top) = p^\top x + q^\top y \leq p^\top \sum_{j=1}^{n} \lambda'_j x(j) + q^\top \sum_{j=1}^{n} \lambda'_j y(j) \leq 0$. Consequently, $H_{p,q,c} \cap T_{\text{CCR}}$ is a facet of T_{CCR}. □

Theorem 4.2.15. *Assume that* $H_{p,q,c} = \{(x, y) \in \mathbb{R}^{m+s} : p^\top x + q^\top y = c\}$ *is calculated by Algorithm* FFC. *If* $c \neq 0$, *then* $H_{p,q,c} \cap T_{\text{CCR}}$ *is not a facet of* T_{CCR}.

Proof. By the definition of T_{CCR}, for each $(x^\top, y^\top)^\top \in T_{\text{CCR}}$ and for all $\alpha > 0$, $\alpha(x^\top, y^\top)^\top \in T_{\text{CCR}}$. At Step 3 of Algorithm FFC, $(p^\top, q^\top)^\top (x(j)^\top, y(j)^\top) \leq c$ $(j = 1, \ldots, n)$. At Step 2 of Algorithm FFC, there exists $i \in \{1, \ldots, n\}$ such that $(p^\top, q^\top)^\top (x(i)^\top, y(i)^\top) = c$. Let $c \neq 0$. Then, there exist $\alpha > 0$ such that $(p^\top, q^\top)^\top \alpha(x(i)^\top, y(i)^\top) > c$. Consequently, $H_{p,q,c} \cap T_{\text{CCR}}$ is not a facet of T_{CCR}. □

Finally, we propose an algorithm to calculate all equations forming $F_{\text{GRS}(L,U)}$. By setting parameters $L \leq 1$ and $U \geq 1$, we can obtain all equations forming the facets of the efficient frontiers of the traditional four models. In order to calculate all equations forming F_{CCR}, we set $L = 0$ and $U > 0$. By using Theorem 4.2.6, we obtain all equations forming F_{CCR}. Similarly, by setting $L = U = 1$, we obtain all equations forming F_{BCC}. If we set $L = 1$ and $U > 1$, then, by using Theorem 4.2.8, we obtain all equations forming F_{IRS}. If we set $L = 0$ and $U = 1$, then all equations forming F_{DRS} are obtained. Obviously, by setting other parameters(for example, $L = 0.5$ and $U = 2$), all equations forming $F_{\text{GRS}(L,U)}$ are obtained. We formulate the algorithm under Theorems 3.3.2, 3.3.3 and 3.3.4.

Algorithm FFG

Step 0

Set $\bar{n} := 2n + m + s$. Moreover, set $P(i)$ $(i = 1, \ldots, 2n)$ and $P'(i)$ $(i = 1, \ldots, \bar{n})$ as follows.

$$P(i) := \begin{cases} L(x(i)^\top, y(i)^\top)^\top & \text{if } i \in \{1, \ldots, n\}, \\ U(x(i-n)^\top, y(i-n)^\top)^\top & \text{if } i \in \{n+1, \ldots, 2n\}. \end{cases} \tag{4.7}$$

$$P'(i) := \begin{cases} P(i) - G & \text{if } i \in \{1, \ldots, 2n\}, \\ e^{i-2n} & \text{if } i \in \{2n+1, \ldots, \bar{n}\}, \end{cases} \tag{4.8}$$

where $G = \frac{1}{2n}(P(1) + \cdots + P(2n))$ and e^j is a vector of $\mathbb{R}^{m+s}$ satisfying $e^j_j = 1$ and $e^j_i = 0$ for each $j \in \{1, \ldots, m+s\}$ and $i \in \{1, \ldots, m+s\} \setminus \{j\}$. Let $c_i := i$ $(i = 1, \ldots, m+s)$. Set $t := 1$ and go to Step 1.

Step 1

If $\dim \{P'(c_i) : i = 1, \ldots, m+s\} = m+s$, then go to Step 2. Otherwise, go to Step 4.

Step 2

Step 2–0

Calculate W by solving the following system of linear equations:

$$\begin{cases} (P'(c_1))^\top W = \alpha(c_1), \\ \qquad \vdots \\ (P'(c_{m+s}))^\top W = \alpha(c_{m+s}), \end{cases}$$

where,

$$\alpha(c_i) := \begin{cases} 1 & \text{if } c_i \in \{1, \ldots, 2n\}, \\ 0 & \text{if } c_i \in \{2n+1, \ldots, \bar{n}\}. \end{cases}$$

Go to Step 3.

Step 2–1

If W calculated at Step 2–0 satisfies the following conditions, then set $V_t := W$ and $t \leftarrow t + 1$.

$$\begin{cases} (P'(j))^\top W \leq 1, \; j = 1, \ldots, 2n, \\ W_i \leq 0, \; i = 1, \ldots, m, \\ W_i \geq 0, \; i = m+1, \ldots, m+s. \end{cases}$$

Otherwise, $\{V_1, \ldots, V_t\}$ remain. If $c_1 = \bar{n} - m - s + 1$, go to Step 4. Otherwise, go to Step 3.

Step 3

Step 3–0

Set $c_{m+s} \leftarrow c_{m+s} + 1$ and $j := m + s$. Go to Step 3–1.

Step 3–1

If $c_j \leq \bar{n} - m - s + j$, set $c_{j'} \leftarrow c_j + j' - j$ for every $j' > j$. Go to Step 1. Otherwise, set $c_{j-1} \leftarrow c_{j-1} + 1, j \leftarrow j - 1$ and go to Step 3–1.

Step 4

For each $i \in \{1, \ldots, t-1\}$, let $(-p_i^\top, q_i^\top)^\top := V_i$, where $p_i \in \mathbb{R}^m$ and $q_i \in \mathbb{R}^s$. For each $i = 1, \ldots, t-1$, let $c_i := 1 + (-p_i^\top, q_i^\top)^\top G$ and the hyperplane forming the efficient frontier is as follows.

$$H_{p_i, q_i, c_i} := \{(x, y) : -p_i^\top x + q_i^\top y = c_i\}.$$

Stop the algorithm.

The algorithm introduced in Section 4.1 calculates some equations forming F_{CCR} by solving many linear programming problems. In this approach, in order to obtain the equations forming the efficient frontiers of the other models, other algorithms must be constructed. In contrast, Algorithm FFG can calculate all equations with respect to all GRS models by setting the parameters. The algorithms proposed in this section obtains all equations forming the efficient frontiers by calculating the vertices of the polar sets based on initial points.

4.3 Formulation for calculating improvements by solving a mixed integer quadratic programming problem

In this section, we illustrate the method proposed by Aparicio, Ruiz and Sirvent [3] to calculate some improvements. We assume that DMU(k) is a CCR-inefficient DMU, that is, $\theta^*_{\mathrm{CCR}}(k) < 1$. By following the improvements, DMU(k) becomes a CCR-efficient DMU satisfying $\theta^*_{\mathrm{CCR}}(k) = 1$.

Let $(\lambda^*, \theta^*_{\mathrm{CCR}}(k))$ be an optimal solution of Problem (CCRD(k)). Then, $\theta^*_{\mathrm{CCR}}(k)$ is one of the traditional improvement for DMU(k). Let DMU($\bar{k}$) $:= (\bar{x}, \bar{y})$ be a DMU defined as follows:

$$\bar{x} := \theta^*_{\mathrm{CCR}}(k)x(k),$$
$$\bar{y} := y(k).$$

Then, $(\lambda^*, 1)$ is a feasible solution of Problem (CCRD($\bar{k}$)). By the optimality of $(\lambda^*, \theta^*_{\mathrm{CCR}}(k))$, the optimal value of Problem (CCRD($\bar{k}$)) equals 1. Hence, $(\bar{x}, \bar{y})$ is

CCR-efficient. This improvement is obtained easily, however it is an difficult improvement since only input values are decreased at the same rate. Therefore, other improvements for inefficient DMUs have been proposed with respect to the CCR or BCC model. For example, Frei and Harker [21] have proposed a least distance projection to F_{CCR} by using the Euclidean norm. Silva, Castro and Thanassoulis [33] have constructed multi-stages procedures for the BCC model. In contrast, Aparicio, Ruiz and Sirvent [3] have proposed a single-stage method by innovating such procedures. The method is formulated as a mixed integer quadratic programming problem for traditional norms to obtain a closest target on F_{CCR} under the following theorem.

Theorem 4.3.1. (Aparicio, Ruiz and Sirvent [3], p.211) *Let $D(k)$ be the set of Pareto-efficient points in T_{CCR} dominating* DMU(k)*. Then, $(x, y) \in D(k) \Leftrightarrow$ there exist $\lambda_j, d_j \geq 0, b_j \in \{0,1\}, j \in E, v_i \geq 1, i = 1, \ldots, m, u_r \geq 1, r = 1, \ldots, s, s_{ik}^- \geq 0, i = 1, \ldots, m$ and $s_{rk}^+ \geq 0, r = 1, \ldots, s$ such that*

$$\begin{aligned}
&x = \sum_{j \in E} \lambda_j x(j), \\
&y = \sum_{j \in E} \lambda_j y(j), \\
&\sum_{j \in E} \lambda_j x(j)_i = x(k)_i - s_{ik}^-, \ i = 1, \ldots, m, \\
&\sum_{j \in E} \lambda_j y(j)_r = y(k)_r + s_{rk}^+, \ r = 1, \ldots, s, \\
&-\sum_{i=1}^{m} v_i x(j)_i + \sum_{r=1}^{s} u_r y(j)_r + d_j = 0, \ j \in E, \\
&d_j \leq \mathrm{M} b_j, \ j \in E, \\
&\lambda_j \leq \mathrm{M}(1 - b_j), \ j \in E,
\end{aligned}$$

where M *is a big positive quantity.*

From above theorem, an improvement for DMU(k) is obtained by solving the following mixed integer quadratic programming problem based on the l_2-norm.

$$
(\mathrm{mADD}(k))\left\{
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{i=1}^{m}(s_{ik}^{-})^2+\sum_{r=1}^{s}(s_{rk}^{+})^2 \\
\text{subject to} & \displaystyle\sum_{j\in E}\lambda_j x(j)_i = x(k)_i - s_{ik}^{-},\ i=1,\ldots,m, \\
 & \displaystyle\sum_{j\in E}\lambda_j y(j)_r = y(k)_r + s_{rk}^{+},\ r=1,\ldots,s, \\
 & \displaystyle -\sum_{i=1}^{m}v_i x(j)_i + \sum_{r=1}^{s}u_r y(j)_r + d_j = 0,\ j\in E, \\
 & v_i \geq 1,\ i=1,\ldots,m, \\
 & u_r \geq 1,\ r=1,\ldots,s, \\
 & d_j \leq \mathrm{M}b_j,\ j\in E, \\
 & \lambda_j \leq \mathrm{M}(1-b_j),\ j\in E, \\
 & b_j \in \{0,1\},\ j\in E, \\
 & d_j \geq 0,\ j\in E, \\
 & \lambda_j \geq 0,\ j\in E, \\
 & s_{ik}^{-} \geq 0,\ i=1,\ldots,m, \\
 & s_{rk}^{+} \geq 0,\ r=1,\ldots,s,
\end{array}\right.
$$

where E is the set of extreme efficient units defined in [13]. Let W_k be the set of all optimizing multipliers for DMU(k). If W_k is not empty, then DMU(k) is said to be DEA-scale-efficient and denote by RE the set of all DEA-scale-efficient DMUs. Then E is defined as follows.

$$E = \{\mathrm{DMU}(k) \in RE : \dim W_k = m+s\}.$$

By changing the objective function depending on a policy of the decision maker and situations, above model can be applied the following cases: the l_1-distance case(to minimize the sum of values of change for all inputs and outputs), the l_∞-distance case(to minimize the maximum value of change for all inputs and outputs), RAM efficiency measure (see [16]) case, SBM efficiency measure (see [39]) case and so on.

4.4 Algorithm for calculating improvements by using the equations forming the efficient frontiers

In this section, we propose four types of improvements for making CCR-inefficient DMUs CCR-efficient (CCR-Pareto-efficient or CCR-weakly-efficient) with a minimal change of input and output values. By many researches, many methods to calculate the closest point over the efficient frontier have been proposed. These methods target the efficient frontier of only one model. In real problems, it is difficult that we select a model to use for evaluation. Therefore, some models are used to evaluate DMUs for a problem. In this thesis, we propose four kinds of improvements as follows. The first improvement is unrestricted, that is, we consider the minimal change of input and output values. The inefficient DMUs can become efficient units by the smallest change under the condition which the improvement targets are feasible. The similar improvements have been proposed as a minimal distance point by many researchers. We introduce a norm to adjust the change of input and output values. However, this improvement is not always possible in the actual situations. Hence, we present the second improvement guaranteeing the feasibility. The second improvement is constrained by the production possibility set of the BCC model. The reason for adding the constraint condition is that the production possibility set of the BCC model can be identified as the feasible region of DMUs. This improvement cannot be obtained by using the method introduced in Section 4.3. We calculate this improvement by utilizing the equations forming the efficient frontiers. Moreover, if a decision maker wants to introduce some conditions for the operation policy, stock status and others, they use the third and fourth improvements. The third improvement is obtained by confining the change of input or output values. By utilizing this improvement, the decision maker can regulate amounts of change of input or output values. The fourth improvement is calculated by according as the order of input and output elements, where the order is provided by the decision maker. Then, the decision maker can control the order of amounts of change of input and output values.

First, we define the norm depending on a symmetric positive definite matrix $A \in \mathbb{R}^{(m+s)\times(m+s)}$ as follows.

$$||Z||_A := \sqrt{Z^\top AZ},\ Z \in \mathbb{R}^{m+s}.$$

Under this norm, we consider the minimal change of input and output values.

Example 4.4.1. In the case of $A = I_{m+s}$, $||\cdot||_A$ corresponds to the Euclidean norm. If A is defined by

$$A = M_k := \begin{pmatrix} \left(\frac{1}{P(k)_1}\right)^2 & & 0 \\ & \ddots & \\ 0 & & \left(\frac{1}{P(k)_{m+s}}\right)^2 \end{pmatrix},$$

then $||\cdot||_A$ means the norm which considered the ratio of input and output values.

We define $d^i(k)(i = 1, \dots, 4)$ as improvements for DMU(k), where each $d^i(k)$ is an optimal solution of Problem (ID$^i(k)$) $(i = 1, \dots, 4)$ formulated as follows:

$$(\mathrm{ID}^i(k)) \begin{cases} \text{minimize} & ||Z||_A \\ \text{subject to} & Z \in B^i(k). \end{cases} \qquad (i = 1, \dots, 4)$$

Here,

$$\begin{aligned} B^1(k) &:= F_{\mathrm{CCR}} - P(k), \\ B^2(k) &:= (F_{\mathrm{CCR}} \cap T_{\mathrm{BCC}}) - P(k), \\ B^3(k) &:= \{Z \in F_{\mathrm{CCR}} - P(k) : \alpha_i \leq Z_i + P(k)_i \leq \beta_i,\ i = 1, \dots, m\}, \\ B^4(k) &:= \{Z \in F_{\mathrm{CCR}} - P(k) : |Z_{t_i}| \leq |Z_{t_{i+1}}|,\ i = 1, \dots, m+s-1\}, \end{aligned}$$

where α_i, $\beta_i \in \mathbb{R}(i = 1, \dots, m)$ are lower and upper limits for the ith element of Z decided by the decision maker of DMU(k) satisfying $\alpha_i \leq P(k)_i \leq \beta_i\,(i = 1, \dots, m)$, $t_i \in \{1, \dots, m+s\}\ (i = 1, \dots m+s)$ satisfy $t_{i'} \neq t_{i''}$ for each $i', i'' \in \{1, \dots, m+s\}\ (i' \neq i'')$. Since $d^1(k)$ solves Problem (ID$^1(k)$), $d^1(k)+P(k)$ has a minimal distance from $P(k)$ over F_{CCR}. The feasible set $B^2(k)$ of Problem (ID$^2(k)$) is the intersection of $B^1(k)$ and T_{BCC}. By confining the feasible set to T_{BCC}, $d^2(k)$ is more realistic than $d^1(k)$. The third improvement $d^3(k)$ is obtained by limiting the amount of change of input values from $P(k)$. Of course, we can limit output values in a similar way. However, if both input and output values are limited, the feasibility of $d^3(k)$ is not guaranteed. Hence, we propose the fourth improvement $d^4(k)$. By deciding $\{t_1, \ldots, t_{m+s}\}$, the decision maker of DMU(k) can control the order of the amount of change of input and output values.

Theorem 4.4.1. *The feasible sets of* $(\mathrm{ID}^i(k))(i = 1, \ldots, 4)$ *are nonempty and closed.*

Proof. First, by the definition of T_{CCR}, $0 \in T_{\mathrm{CCR}}$. Since T_{CCR} is closed, by Theorem 4.2.4, F_{CCR} is closed and $0 \in F_{\mathrm{CCR}}$. Hence, $Z = -P(k)$ is a feasible solution and $\{Z - P(k) : Z \in F_{\mathrm{CCR}}\}$ is closed. Therefore, the feasible set of $(\mathrm{ID}^1(k))$ is nonempty and closed.

Second, for each DMU$(j) \in F_{\mathrm{CCR}}$, DMU$(j) \in T_{\mathrm{BCC}}$. Since T_{BCC} is closed, $F_{\mathrm{CCR}} \cap T_{\mathrm{BCC}}$ is nonempty and closed. Let DMU$(j') \in F_{\mathrm{CCR}}$, then $Z = -P(j')$ is a feasible solution and $\{Z - P(k) : Z \in F_{\mathrm{CCR}} \cap T_{\mathrm{BCC}}\}$ is closed. Hence, the feasible set of $(\mathrm{ID}^2(k))$ is nonempty and closed.

Third, we note that $(-p_j^\top, q_j^\top)^\top(\alpha_1, \ldots, \alpha_m, 0, \ldots, 0) < 0$ for each $j \in S_c$. Let $j' \in \arg\max\left\{\frac{-p_j^\top(\alpha_1,\ldots,\alpha_m)}{q_{1,j}} : j \in S_c\right\}$ and set $\gamma := \frac{p_{j'}^\top(\alpha_1,\ldots,\alpha_m)}{q_{1,j'}}$. Then, we obtain $(-p_j^\top, q_j^\top)^\top(\alpha_1, \ldots, \alpha_m, \gamma, 0, \ldots, 0) \leq 0$ for each $j \in S_c$ and by the definitions of j' and γ, $(-p_{j'}^\top, q_{j'}^\top)^\top(\alpha_1, \ldots, \alpha_m, \gamma, 0, \ldots, 0) = 0$. By Theorem 4.2.4, $(\alpha_1 - P(k)_1, \ldots, \alpha_m - P(k)_m, \gamma - P(k)_{m+1}, -P(k)_{m+s}, \ldots, -P(k)_{m+s}) \in F_{\mathrm{CCR}} - P(k)$. Therefore, the feasible set of $(\mathrm{ID}^3(k))$ is nonempty and closed.

Fourth, since $P(k) \notin F_{\mathrm{CCR}}$, we obtain $(-p_j^\top, q_j^\top)^\top(P(k)_1, \ldots, P(k)_{m+s}) < 0$ for each $j \in S_c$. Let $j' \in \arg\max\left\{\frac{(p_j^\top, -q_j^\top)^\top(P(k)_1,\ldots,P(k)_{m+s})}{-p_{1,j}-\cdots-p_{m,j}+q_{1,j}+\cdots+q_{s,j}} : j \in S_c\right\}$ and $\alpha := \frac{(p_{j'}^\top, -q_{j'}^\top)^\top(P(k)_1,\ldots,P(k)_{m+s})}{-p_{1,j'}-\cdots-p_{m,j'}+q_{1,j'}+\cdots+q_{s,j'}}$. Then, $(-p_j^\top, q_j^\top)^\top(P(k)_1 - \alpha, \ldots, P(k)_m - \alpha, P(k)_{m+1} + \alpha, \ldots, P(k)_{m+s} + \alpha) \leq 0$ for each $j \in S_c$ and $(-p_{j'}^\top, q_{j'}^\top)^\top(P(k)_1 - \alpha, \ldots, P(k)_m - \alpha, P(k)_{m+1} + \alpha, \ldots, P(k)_{m+s} + \alpha) = 0$. By Theorem 4.2.4, $(P(k)_1 - \alpha, \ldots, P(k)_m - \alpha, P(k)_{m+1} + \alpha, \ldots, P(k)_{m+s} + \alpha) \in F_{\mathrm{CCR}}$. Obviously, $(P(k)_1 - \alpha, \ldots, P(k)_m - \alpha, P(k)_{m+1}+\alpha, \ldots, P(k)_{m+s}+\alpha) \in \{Z \in \mathbb{R}^{m+s} : |Z_{t_q}| \leq |Z_{t_{q+1}}|, q = 1, \ldots, m+s-1\}$. Hence, $F_{\mathrm{CCR}} \cap \{Z \in \mathbb{R}^{m+s} : |Z_{t_q}| \leq |Z_{t_{q+1}}|, q = 1, \ldots, m+s-1\}$ is nonempty. Therefore, the feasible set of $(\mathrm{ID}^4(k))$ is nonempty and closed. □

We propose the following algorithm for obtaining four types of improvements $d^i(k)\,(i \in \{1, \ldots, 4\})$. Let N_c be the number of elements of S_c. improvements for

DMU(k) are obtained by the following algorithm:

Algorithm ICCR

Step 0

Select $i \in \{1, \dots, 4\}$ (Choose the type of the improvement). Set $j := 1$ and go to Step 1.

Step 1

Let $d^i_j(k)$ be an optimal solution of Problem $(\mathrm{ID}^i_j(k))$ defined as follows:

$$(\mathrm{ID}^i_j(k)) \left\{ \begin{array}{ll} \text{minimize} & ||Z||_A \\ \text{subject to} & Z \in B^i_j(k), \end{array} \right.$$

where

$$\begin{aligned} B^1_j(k) &:= \{Z : (Z + P(k))^\top W_j = 0\}, \\ B^2_j(k) &:= \{Z : (Z + P(k))^\top W_j = 0, (Z + P(k))^\top W_l \leq c_l \text{ for each } l \in S_b\}, \\ B^3_j(k) &:= \{Z : (Z + P(k))^\top W_j = 0, \alpha_l \leq Z_l + P(k)_l \leq \beta_l,\ l = 1, \dots, m\}, \\ B^4_j(k) &:= \{Z : (Z + P(k))^\top W_j = 0, |Z_{t_l}| \leq |Z_{t_{l+1}}|,\ l = 1, \dots, m + s - 1\}. \end{aligned}$$

If $j = N_c$, then go to Step 2. Otherwise, set $j \leftarrow j + 1$ and go to Step 1.

Step 2

Select $j' \in \arg\min\{||d^i_j(k)||_A : j \in S_c\}$ and set $d^i(k) := d^i_{j'}(k)$. This algorithm terminates.

We can execute Algorithm ICCR using the existing nonlinear optimization techniques (e.g. [6]). The existence and properties of an optimal solution are proved by the following theorems.

Theorem 4.4.2. *For each $i \in \{1, \dots, 4\}$, Problem $(\mathrm{ID}^i_j(k))$ has an optimal solution.*

Proof. By Theorem 4.4.1, for each $i \in \{1, \dots, 4\}$, $B^i_j(k)$ is nonempty and closed. Since $B^i_j(k)$ is nonempty, for each $(x', y') \in B^i_j(k)$, $\bar{B}^i_j(k) := B^i_j(k) \cap \{(x^\top, y^\top)^\top : ||(x^\top, y^\top)^\top||_A \leq ||(x'^\top, y'^\top)^\top||_A\}$ is compact. Therefore, Problem $(\mathrm{ID}^i_j(k))$ is equivalent to the following problem.

$$(\overline{\mathrm{ID}}^i_j(k)) \left\{ \begin{array}{ll} \text{minimize} & ||Z||_A \\ \text{subject to} & Z \in \bar{B}^i_j(k), \end{array} \right.$$

Since the objective function is continuous and the feasible set is compact, Problem $(\overline{\mathrm{ID}}_j^i(k))$ has an optimal solution. By the definition of $\bar{B}_j^i(k)$, an optimal solution of Problem $(\overline{\mathrm{ID}}_j^i(k))$ is also an optimal solution of Problem $(\mathrm{ID}_j^i(k))$. Therefore, Problem $(\mathrm{ID}_j^i(k))$ has an optimal solution. □

We note that Problem $(\mathrm{ID}_j^i(k))$ is a standard quadratic programming problem. Since $N_c < \infty$, Algorithm ICCR terminates within a finite number of iterations.

Theorem 4.4.3. *For each CCR-inefficient* DMU(k)*, let* $d^i(k)$ $(i \in \{1, \dots, 4\})$ *be an optimal solution calculated by Algorithm* ICCR*. Then,* $P(k) + d^i(k) \in F_{\mathrm{CCR}}$.

Proof. Let W_j $(j \in S_c)$ be all normal vectors calculated by Algorithm FFA. In order to obtain a contradiction, we suppose that $P(k)+d^i(k) \notin F_{\mathrm{CCR}}$. By Theorem 4.2.4, $P(k) + d^i(k) \notin T_{\mathrm{CCR}}$, and by Theorem 4.2.2, there exists $j \in S_c$ such that $(P(k) + d^i(k))^\top W_j > 0$. Since $P(k) \in \operatorname{int} T_{\mathrm{CCR}}$, from Theorem 4.2.2, $P(k)^\top W_j < 0$ and $(\alpha(P(k) + d^i(k)) + (1 - \alpha)P(k))^\top W_j = (P(k) + \alpha d^i(k))^\top W_j = 0$, where $\alpha := -\frac{P(k)^\top W_j}{d^i(k)^\top W_j}$. Since $0 < \alpha < 1$ and $d^i(k)$ satisfies the additional conditions of $B_j^i(k)$ $(i = 2, 3, 4)$, $\alpha d^i(k)$ also satisfies the additional conditions. Therefore, $\alpha d^i(k)$ is a feasible solution of Problem $(\mathrm{ID}_j^i(k))$. By the definition of $d_j^i(k)$, we have the following inequality: $||d_j^i(k)||_A \leq ||\alpha d^i(k)||_A < ||d^i(k)||_A$. This contradicts the optimality of $d^i(k)$ for Algorithm ICCR. Consequently, $P(k) + d^i(k) \in F_{\mathrm{CCR}}$. □

If the decision maker of DMU(k) wants to obtain a CCR-Pareto-efficient point, that is, an optimal slackness is zero, then he can modify Algorithm ICCR as follows.

By replacing S_c, N_c and $B_j^i(k)\,(i = 1, \dots, 4)$ in Algorithm ICCR by $S_c^{'}$, $N_c^{'}$ and $B_j^{i'}(k)\,(i = 1, \dots, 4)$ as follows.

$$S_c^{'} := \{j \in S_c : W_{i,j} \neq 0,\, i = 1, \dots, m+s\},$$

where W_j $(j \in S_c)$ are all vectors calculated by Algorithm FFA. Let $N_c^{'}$ be the number of elements of $S_c^{'}$.

$$B_j^{1'}(k) := \{Z : (Z + P(k))^\top W_j = 0,\, (Z + P(k))^\top W_o \leq 0 \text{ for each } o \in S_c \setminus S_c^{'}\},$$

$$B_j^{2'}(k) := \{Z : (Z + P(k))^\top W_j = 0,\, (Z + P(k))^\top W_o \leq 0 \text{ for each } o \in S_c \setminus S_c^{'},$$
$$(Z + P(k))^\top W_l \leq c_l \text{ for each } l \in S_b\},$$

$$B_j^{3'}(k) := \{Z : (Z + P(k))^\top W_j = 0,\ (Z + P(k))^\top W_o \le 0 \text{ for each } o \in S_c \setminus S_c^{'},$$
$$\alpha_l \le Z_l + P(k)_l \le \beta_l,\ l = 1, \dots, m\},$$
$$B_j^{4'}(k) := \{Z : (Z + P(k))^\top W_j = 0,\ (Z + P(k))^\top W_o \le 0 \text{ for each } o \in S_c \setminus S_c^{'},$$
$$|Z_{t_l}| \le |Z_{t_{l+1}}|,\ l = 1, \dots, m + s - 1\}.$$

Moreover, we assume that $B_j^{i'}(k)\,(i = 1, \dots, 4)$ are nonempty. Then we can obtain a CCR-Pareto-efficient point.

Theorem 4.4.4. *For each CCR-inefficient* DMU(k)*, let* $d^i(k)$ $(i \in \{1, \dots, 4\})$ *be an optimal solution calculated by modified Algorithm* ICCR. *Then,* $P(k)+d^i(k) \in F_{\mathrm{CCR}}$ *is a CCR-Pareto-efficient point.*

Proof. We can prove the existence of an optimal solution and $P(k) + d^i(k) \in F_{\mathrm{CCR}}$ in a way similar to Theorems 4.4.2 and 4.4.3. In order to obtain a contradiction, we suppose that $P(k)+d^i(k)$ has positive slack, that is, there exist slack vectors $s^x \ge 0 \in \mathbb{R}^m$ and $s^y \ge 0 \in \mathbb{R}^s$ satisfying $(s^{x\top}, s^{y\top}) \ne (0, 0)$, and $P(k)+d^i(k)+(-s^{x\top}, s^{y\top})^\top \in F_{\mathrm{CCR}}$. Since $d^i(k)$ is an optimal solution, there exists $j \in S_c^{'}$ such that $(d^i(k) + P(k))^\top W_j = 0$. Then $(d^i(k) + P(k) + (-s^{x\top}, s^{y\top})^\top)^\top W_j = (-s^{x\top}, s^{y\top}) W_j > 0$. By Theorems 4.2.2 and 4.2.4, this contradicts $P(k) + d^i(k) + (-s^{x\top}, s^{y\top})^\top \in F_{\mathrm{CCR}}$. Therefore, $P(k) + d^i(k)$ is a CCR-Pareto-efficient point. □

By introducing a parameter α, we propose an algorithm to calculate a minimal distance point or a Pareto-efficient point on F_{CCR} as an improvement. The improvement of DMU(k) is calculated as follows:

Algorithm GIT

Step 0

Select $\alpha \in \{0, 1\}$ (Choose the type of the improvment). Set $j := 1$ and go to Step 1.

Step 1

If $\alpha = 1$, then set

$$S_c^{'} := \{l \in S_c : W_{li} \ne 0,\ i = 1, \dots, m + s\} \text{ and } S := S_c^{'}.$$

If $\alpha = 0$, then set

$$S := S_c.$$

Let N be the number of elements of S. Go to Step 2.

Step 2

Let $d_j^\alpha(k)$ be an optimal solution of Problem $(\mathrm{MIT}_j^\alpha(k))$ defined as follows:

$$(\mathrm{MIT}_j^\alpha(k)) \begin{cases} \text{minimize} & ||Z||_A \\ \text{subject to} & (Z + P(k))^\top W_j = 0, \\ & \alpha(Z + P(k))^\top W_o \leq 0 \text{ for each } o \in S, \end{cases}$$

where j denote the jth element of S. If $j = N$, then go to Step 3. Otherwise, set $j \leftarrow j + 1$ and go to Step 2.

Step 3

Select $j^{'} \in \arg\min\{||d_j^\alpha(k)||_A : j \in S\}$ and set $d^\alpha(k) := d_{j^{'}}^\alpha(k)$. This algorithm terminates.

We obtain a minimal distance point or a Pareto-efficient point based on parameter α as indicated by the following theorems.

Theorem 4.4.5. *For each CCR-inefficient* DMU(k), *let* $d^\alpha(k)$ $(\alpha \in \{0, 1\})$ *be an optimal solution calculated by Algorithm* GIT. *Then,* $P(k) + d^\alpha(k) \in F_{\mathrm{CCR}}$.

Proof. We prove the case of $\alpha = 0$. In order to obtain a contradiction, we suppose that $P(k) + d^0(k) \notin F_{\mathrm{CCR}}$. By Theorem 4.2.4, $P(k) + d^0(k) \notin T_{\mathrm{CCR}}$, and by Theorem 4.2.2, there exists $j \in S_c$ such that $(P(k) + d^0(k))^\top W_j > 0$. Since DMU$(k)$ is a CCR-inefficient DMU, $P(k) \in \operatorname{int} T_{\mathrm{CCR}}$. Hence, from Theorem 4.2.2, $P(k)^\top W_j < 0$ and $(\gamma(P(k) + d^0(k)) + (1 - \gamma)P(k))^\top W_j = (P(k) + \gamma d^0(k))^\top W_j = 0$, where $\gamma := -\frac{P(k)^\top W_j}{d^0(k)^\top W_j}$. Since $(P(k) + d^0(k))^\top W_j > 0$, we obtain $0 < \gamma < 1$. Therefore, $\gamma d^0(k)$ is a feasible solution of Problem $(\mathrm{MIT}_j^0(k))$. By the definition of $d_j^0(k)$, we have the following inequality: $||d_j^0(k)||_A \leq ||\gamma d^0(k)||_A < ||d^0(k)||_A$. This contradicts the optimality of $d^0(k)$ for Algorithm GIT. Consequently, $P(k) + d^0(k) \in F_{\mathrm{CCR}}$. For the case of $\alpha = 1$, we replace S_c by $S_c^{'}$ and can complete the proof in a way similar to the case of $\alpha = 0$. □

By Theorem 4.4.5, we note that $P(k) + d^0(k)$ is a CCR-efficient point for each CCR-inefficient DMU(k). Moreover, we obtain a Pareto-efficient point based on parameter $\alpha = 1$ as indicated by the following theorem.

Theorem 4.4.6. *For each CCR-inefficient* DMU(k)*, let* $d^1(k)$ *be an optimal solution calculated by modified Algorithm* GIT $(\alpha = 1)$*. Then,* $P(k) + d^1(k) \in F_{\text{CCR}}$ *is a CCR-Pareto-efficient point.*

Proof. The existence of an optimal solution and $P(k) + d^1(k) \in F_{\text{CCR}}$ are proved by a way similar to Theorems 4.4.2 and 4.4.3. In order to obtain a contradiction, we suppose that $P(k) + d^1(k)$ has positive slack, that is, there exist slack vectors $s^x \geq 0 \in \mathbb{R}^m$ and $s^y \geq 0 \in \mathbb{R}^s$ satisfying $(s^{x\top}, s^{y\top}) \neq (0, 0)$, and $P(k) + d^1(k) + (-s^{x\top}, s^{y\top})^\top \in F_{\text{CCR}}$. Since $d^1(k)$ is an optimal solution of Problem $(\text{MIT}_j^1(k))$ for some $j \in \{1, \ldots, N\}$, there exists $j \in S$ such that $(d^1(k) + P(k))^\top W_j = 0$. Then $(d^1(k) + P(k) + (-s^{x\top}, s^{y\top})^\top)^\top W_j = (-s^{x\top}, s^{y\top}) W_j > 0$. By Theorems 4.2.2 and 4.2.4, this contradicts $P(k) + d^1(k) + (-s^{x\top}, s^{y\top})^\top \in F_{\text{CCR}}$. Therefore, $P(k) + d^1(k)$ is a CCR-Pareto-efficient point. □

Theorem 4.4.7. *For each CCR-inefficient* DMU(k)*, let* $d^0(k)$ *and* $d^1(k)$ *be optimal solutions calculated by modified Algorithm* GIT $(\alpha = 0)$ *and* $(\alpha = 1)$*, respectively. Then, for each* $\lambda \in (0, 1)$*,* $d^\lambda(k) := \lambda(P(k) + d^0(k)) + (1 - \lambda)(P(k) + d^1(k)) \in T_{\text{CCR}}$*.*

Proof. By Theorems 4.2.4 and 4.4.5, $P(k) + d^0(k)$ and $P(k) + d^1(k)$ are contained in T_{CCR}. Since T_{CCR} is a closed convex set, $d^\lambda(k) := \lambda(P(k) + d^0(k)) + (1 - \lambda)(P(k) + d^1(k)) \in T_{\text{CCR}}$ for each $\lambda \in (0, 1)$. □

We note that $d^\lambda(k)$ is not always contained in F_{CCR}, since F_{CCR} is not convex set. In order to calculate a point on F_{CCR} based on $d^\lambda(k)$, we consider a projection. Let $\bar{\beta} := \min\{\beta : (P(k) + \beta(d^\lambda(k) - P(k)))^\top W_j = 0 \text{ for some } j \in S_c\}$. Then, by Theorems 4.2.2 and 4.2.4, $P(k) + \bar{\beta}(d^\lambda(k) - P(k)) \in F_{\text{CCR}}$. We propose this point $P(k) + \bar{\beta}(d^\lambda(k) - P(k))$ as improvement intermediate between the two improvements which are obtained based on $d^0(k)$ and $d^1(k)$.

In previous method introduced in Section 4.3 to calculate an improvement, the mixed integer linear programming problems for typical norms were formulated. In this section, we have proposed the algorithms by using the equations forming the efficient frontiers. By using this approach, we can first obtain an improvement based on more than two models. Moreover, we have added additional constraint conditions into the previous improvement to deal with various situations.

4.5 Example

Now, we perform a numerical analysis for 10 Japanese banks by utilizing algorithms provided in this thesis. As shown in Table 4.4, each bank has the ordinary profit as the single output. The number of employees and total assets are the two inputs used to generate the output.

Table 4.4: Inputs and Output values for 10 Japanese banks, 2008

Bank	Input 1 (persons)	Input 2 (one hundred million Japanese yen)	Output (one hundred million Japanese yen)
B1	3701	119895	3179
B2	3675	98359	2688
B3	3659	80955	2180
B4	3004	59600	1563
B5	2887	66373	1477
B6	2872	90984	2450
B7	2752	60770	1852
B8	2506	49008	1137
B9	2268	41151	1148
B10	2148	41158	1124

The efficiency scores and the RTS are shown in the Table 4.5. All efficiency scores are calculated by using Theorems 4.2.11 and 4.2.12 and the RTS are obtained by using Theorem 4.2.13.

Three banks are CCR-efficient and they do not have to think the improvement. Another bank's improvements are given in Tables 4.6-4.9. The improvement over an efficient frontier of CCR model ($A = A_k$) is shown in Table 4.6. Improvements contained in T_{BCC}, T_{IRS} and T_{DRS} ($A = A_k$) are given in Tables 4.7, 4.8 and 4.9, respectively. The improvement over an efficient frontier of CCR model think decreasing inputs and increasing outputs. In contrast, other improvements might increasing

Table 4.5: DEA analysis for 10 Japanese banks, 2008

Bank	CCR	BCC	IRS	DRS	RTS
B1	1.000000	1.000000	1.000000	1.000000	C
B2	0.961359	0.996536	0.961359	0.996536	-
B3	0.884268	0.931186	0.884268	0.931186	-
B4	0.860520	0.884500	0.884500	0.860520	-
B5	0.741447	0.814268	0.814268	0.741447	-
B6	1.000000	1.000000	1.000000	1.000000	C
B7	1.000000	1.000000	1.000000	1.000000	C
B8	0.761275	0.859975	0.859975	0.761275	-
B9	0.915398	1.000000	1.000000	0.915398	I
B10	0.896108	1.000000	1.000000	0.896108	I

inputs or decreasing outputs. This means that the DMU is impossible to become CCR-efficient in the PPS of the other models by decreasing inputs. Similarly, the DMU is impossible to become CCR-efficient in the PPS of the other models by increasing outputs.

Table 4.6: Improvement over F_{CCR} $(A = A_k)$

Bank	Input 1	Input 2	Output
B1	-	-	-
B2	-43.28	-358.63	90.72
B3	0.00	-5290.05	125.93
B4	0.00	-4769.59	107.99
B5	0.00	-11664.62	190.27
B6	-	-	-
B7	-	-	-
B8	0.00	-7415.07	130.57
B9	0.00	-1895.63	48.33
B10	0.00	-2368.56	58.13

Table 4.7: Improvement contained in T_{BCC} $(A = A_k)$

Bank	Input 1	Input 2	Output
B1	-	-	-
B2	-802.17	-7345.86	-237.27
B3	-789.48	-20185.00	-328.00
B4	-91.89	1170.00	289.00
B5	27.46	-5603.00	375.00
B6	-	-	-
B7	-	-	-
B8	246.09	11785.65	715.47
B9	637.19	19619.00	704.00
B10	604.85	19826.76	732.25

Table 4.8: Improvement contained in T_{IRS} $(A = A_k)$

Bank	Input 1	Input 2	Output
B1	-	-	-
B2	-782.55	-6663.43	-220.03
B3	-766.14	10754.95	288.33
B4	-246.99	1175.20	289.16
B5	216.46	2166.65	611.68
B6	-	-	-
B7	-	-	-
B8	276.21	11833.95	717.19
B9	485.72	19635.57	704.51
B10	613.94	19628.56	728.51

Table 4.9: Improvement contained in T_{DRS} $(A = A_k)$

Bank	Input 1	Input 2	Output
B1	-	-	-
B2	-1196.07	-17582.18	-558.71
B3	-1203.75	-1025.48	-71.04
B4	-1347.67	-7127.44	-150.04
B5	-1256.57	-14720.64	-86.13
B6	-	-	-
B7	-	-	-
B8	-874.90	2665.70	254.44
B9	-644.53	10281.06	236.93
B10	-555.89	9280.97	234.19

Chapter 5

CROSS EFFICIENCY EVALUATION

In the CCR model, each DMU is evaluated by an advantageous weight. In the cross efficiency evaluation, each DMU is evaluated by using the most advantageous weights for all DMUs. Then, we can evaluate all DMUs as a linear-order relation having the dominance relationships for all DMUs. Therefore, the cross efficiency evaluation has been recommended as an alternative methodology for ranking DMUs in DEA [32]. In Section 5.1, we introduce some basic cross efficiency evaluation methods. In Section 5.2, we formulate new methods of the cross efficiency evaluation by using the equations forming F_{CCR}. In Section 5.3, we propose two kinds of other evaluation methods by using the equations forming F_{CCR}. In Section 5.4, we show a numerical experiment to compare the cross efficiency scores of evaluation methods in Sections 5.1 and 5.2.

5.1 Basic cross efficiency evaluations

In order to calculate the cross efficiency scores for all DMUs, the optimal solutions of the CCR model for all DMUs are used. We obtain an optimal solution $(v^*(k),\, u^*(k))$ by solving Problem (CCRLP(k)) for each DMU(k) $(k = 1, \ldots, n)$. By using $(v^*(i),\, u^*(i))$, we calculate the cross efficiency score of DMU(k) for DMU(i) as follows:

$$\theta_i^*(k) := \frac{u^*(i)^\top y(k)}{v^*(i)^\top x(k)}$$

We note that $0 < \theta_i^*(k) \leq 1$ for each $i, k \in \{1, \ldots, n\}$ by the constraint conditions of Problem (CCRLP(k)). This value means that DMU(k) is evaluated under an advantageous weight for DMU(i). Then, DMUs are ranked according to the average of the cross efficiency scores for all DMUs shown in Table 5.1. In Table 5.1, we obtain an $n \times n$ matrix and the average of all elements of the kth row means the cross efficiency score of DMU(k).

Table 5.1: Cross efficiency of n DMUs

DMU	Target DMU				Average cross efficiency
	1	2	$\cdots$	n	
1	$\theta_1^*(1)$	$\theta_2^*(1)$	$\cdots$	$\theta_n^*(1)$	$\frac{1}{n}\sum_{j=1}^n \theta_j^*(1)$
2	$\theta_1^*(2)$	$\theta_2^*(2)$	$\cdots$	$\theta_n^*(2)$	$\frac{1}{n}\sum_{j=1}^n \theta_j^*(2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$\theta_1^*(n)$	$\theta_2^*(n)$	$\cdots$	$\theta_n^*(n)$	$\frac{1}{n}\sum_{j=1}^n \theta_j^*(n)$

In general, the CCR model has many optimal solutions. Then, it is not necessary that the cross efficiency scores for all optimal solutions coincide. In other words, the ranking of DMUs may differ depending on the solution method. To resolve this problem, Sexton, Silkman and Hogan considered the following problem to decide a weight to achieve the intended objective of minimizing the average of cross efficiency scores of the other DMUs under the condition that gives the maximum efficiency score for the object DMU in [32].

$$
(\mathrm{AVE}(k)) \left\{
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{j=1, j\neq k}^{n} \frac{u^\top y(j)}{v^\top x(j)} \\
\text{subject to} & \displaystyle\frac{u^\top y(k)}{v^\top x(k)} = \theta^*(k), \\
& \displaystyle\frac{u^\top y(j)}{v^\top x(j)} \leq 1, \; j = 1, \ldots, n; j \neq k, \\
& u \geq 0, \\
& v \geq 0.
\end{array}
\right.
$$

By solving Problem (AVE(k)), DMU(k) obtains a weight which is relatively highly-regarded compared to the other DMUs. However, since Problem (AVE(k)) is to minimize the sum of $n-1$ linear fractional functions, it is difficult to solve Problem (AVE(k)). Therefore, many researchers have proposed secondary goal approaches to avoid the facultativity of the cross efficiency evaluation by formulating other problems. Sexton, Silkman and Hogan have formulated the following problem instead of solving Problem (AVE(k)).

$$
(\mathrm{SSH}(k))\left\{\begin{array}{ll}
\text{minimize} & u^\top\left(\sum\limits_{j=1,j\neq k}^{n} y(j)\right) - v^\top\left(\sum\limits_{j=1,j\neq k}^{n} x(j)\right)\\
\text{subject to} & \dfrac{u^\top y(k)}{v^\top x(k)} = \theta^*(k),\\
& \dfrac{u^\top y(j)}{v^\top x(j)} \leq 1,\ j=1,\dots,n; j\neq k,\\
& v^\top x(k) = 1,\\
& u \geq 0,\\
& v \geq 0.
\end{array}\right.
$$

On the other hand, Doyle and Green [18] have proposed another problem which is the so-called aggressive formulation. This problem is one of the most commonly used secondary goal approach since the score based on the aggressive formulation is closer than the score by solving Problem (SSH(k)) (see [18]). By using an optimal solution of the following problem, the aggressive cross efficiency score is calculated.

$$
(\mathrm{AGG}(k))\left\{\begin{array}{ll}
\text{minimize} & \dfrac{u^\top\left(\sum\limits_{j=1,j\neq k}^{n} y(j)\right)}{v^\top\left(\sum\limits_{j=1,j\neq k}^{n} x(j)\right)}\\
\text{subject to} & \dfrac{u^\top y(k)}{v^\top x(k)} = \theta^*(k),\\
& \dfrac{u^\top y(j)}{v^\top x(j)} \leq 1,\ j=1,\dots,n; j\neq k,\\
& u \geq 0,\\
& v \geq 0.
\end{array}\right.
$$

Problem (AGG(k)) is transformed into the linear programming problem by limiting the denominator of the objective function equals 1:

$$
(\mathrm{AGGLP}(k)) \begin{cases} \text{minimize} & u^\top \left(\sum_{j=1, j \neq k}^{n} y(j) \right) \\ \text{subject to} & v^\top \left(\sum_{j=1, j \neq k}^{n} x(j) \right) = 1, \\ & u^\top y(k) - \theta^*(k) v^\top x(k) = 0, \\ & u^\top y(j) - v^\top x(j) \leq 0,\ j = 1, \ldots, n; j \neq k, \\ & u \geq 0, \\ & v \geq 0. \end{cases}
$$

By solving Problem (AGGLP(k)), an advantageous weight for DMU(k) is obtained. The aim of this problem is to minimize the cross efficiency scores of the other DMUs by solving a linear programming problem. In this thesis, we improve the aggressive formulation by considering a sum fractional programming problem.

Moreover, in order to determine the cross efficiency score uniquely, the modified cross efficiency evaluation has been proposed in [22]. The method calculates upper and lower bounds of cross efficiency score. By using the two scores, a cross efficiency score is calculated based on seven criterions in [22]. The problem which calculates a weight to obtain the lower bound of cross efficiency score of DMU(l) for target DMU(k) is formulated as follows:

$$
(\mathrm{MMIN}(k)) \begin{cases} \text{minimize} & \dfrac{u^\top y(l)}{v^\top x(l)} & \\ \text{subject to} & \dfrac{u^\top y(j)}{v^\top x(j)} \leq 1,\ j = 1, \ldots, n, & (8) \\ & \dfrac{u^\top y(k)}{v^\top x(k)} = \theta^*(k), & (9) \\ & u \geq 0, & (10) \\ & v \geq 0. & (11) \end{cases}
$$

Moreover, the problem for calculating a weight to obtain the upper bound of cross efficiency score of DMU(l) for target DMU(k) is formulated as follows:

$$
(\mathrm{MMAX}(k)) \begin{cases} \text{maximize} & \dfrac{u^\top y(l)}{v^\top x(l)} \\ \text{subject to} & (8), \ldots, (11). \end{cases}
$$

5.2 Cross efficiency evaluations by using the facets of F_{CCR}

In this section, we propose three kinds of evaluation methods utilizing the facets of F_{CCR}. In the case where $m + s \geq 3$, the efficient frontier is formed by some facets. Let h be the number of facets forming the efficient frontier of the CCR model. Then, $\bar{\theta}_j(k)$ which is a efficiency score of DMU(k) based on the jth facet is calculated as follows:

$$\bar{\theta}_j(k) := \frac{q_j^\top y(k)}{p_j^\top x(k)},$$

where $H_j := \{(x, y) : -p_j^\top x + q_j^\top y = 0\}$ $(p_j \geq 0, q_j \geq 0)$ is the jth facet forming the efficient frontier of the CCR model. By using the equations forming F_{CCR}, we can calculate optimal solutions of Problems (AGGLP(k)), (MMIN(k)) and (MMAX(k)). Moreover, we examine the primary objective of minimizing the average of cross efficiency scores of the other DMUs easily.

First, we suggest *minimal facet cross efficiency evaluation method.* In this method, for each combination of object DMU(l) and target DMU(k), we decide the facet which gives the minimum score for DMU(l) and gives the maximum score for DMU(k) in the CCR model as follows:

$$j' \in \arg\min \left\{ \frac{q_j^\top y(l)}{p_j^\top x(l)} : \bar{\theta}_j(k) = \theta^*(k),\, j = 1, \ldots, h \right\}$$

We select only one facet and the j'th facet is used to calculate a cross efficiency score for each DMU. This means that target DMU(k) may select different facet for each object DMU. The aim of this method is to obtain the lower bound of the cross efficiency score. The score of this method coincides with the optimal value of Problems (MMIN(k)).

Second, we suggest *aggressive facet cross efficiency evaluation method.* For each target DMU(k), we decide a facet which minimizes the value of aggressive object function in [18] and gives the maximum score for DMU(k) as follows:

$$j^{'} \in \arg\min \left\{ \frac{q_j^\top \left(\sum\limits_{l=1,l\neq k}^{n} y(l) \right)}{p_j^\top \left(\sum\limits_{l=1,l\neq k}^{n} x(l) \right)} : \bar{\theta}_j(k) = \theta^*(k),\, j = 1, \ldots, h \right\}$$

Based on the idea of the traditional cross efficiency evaluation, we consider that target DMU(k) selects the same facet for other DMUs. The score of this method coincides with the optimal value of Problems (AGG(k)). We can prove this equivalence relation in a way similar to Theorem 4.2.11.

Third, we propose *sum minimal facet cross efficiency evaluation method* which is a new evaluation method based on the primary objective in the cross efficiency evaluation. For each target DMU(k), we decide a facet which minimizes the sum of efficiency scores of the other DMUs and gives the maximum score for DMU(k) as follows:

$$j^{'} \in \arg\min \left\{ \sum_{l=1,l\neq k}^{n} \frac{q_j^\top y(l)}{p_j^\top x(l)} : \bar{\theta}_j(k) = \theta^*(k),\, j = 1, \ldots, h \right\}$$

The aim of this method is to select a facet which minimizes the average of efficiency scores of the other DMUs. So far, since Problem (AVE(k)) is not solved easily, we have compromised in Problem(AGG(k)) for minimizing the average score of the other DMUs. By using this method, we can obtain a closer cross efficiency score to Problem (AVE(k)) than the score by solving Problem (AGGLP(k)) shown in Table 5.2. In order to examine the relationship between the aggressive facet cross efficiency evaluation and the sum minimal facet cross efficiency evaluation, we perform experiments with 20 test problems for each situations. The input and output values are randomly-determined. For each problem, we decide facets for all DMUs based on the two methods. When each DMU selects the same facet and at least one DMU selects different facets in the two methods, the two methods are called match and mismatch, respectively. If the facets match, then it means that our method proposed in this thesis obtains the same score as the traditional secondary goal approach. In contrast, mismatch of the facets means that we obtain a closer cross efficiency score to the original goal than the score by solving the traditional formulation. As the numbers of input, output and DMU increase, we can obtain a weight reducing the sum of cross efficiency scores of the other DMUs than the optimal solution of Problem (AGGLP(k)) frequently.

Table 5.2: The ratios of match and mismatch

Situation (20 DMUs)	mismatch	match
2 inputs and 1 output	0.1	0.9
2 inputs and 2 outputs	0.25	0.75
3 inputs and 3 outputs	0.9	0.1
Situation (50 DMUs)	**mismatch**	**match**
2 inputs and 1 output	0.25	0.75
2 inputs and 2 outputs	0.65	0.35
3 inputs and 3 outputs	1.0	0

5.3 Weighted sum evaluation

In this section, we propose two evaluation methods to evaluate DMUs by calculating the weighted sum of the scores obtained based on the equations forming the facets of the efficient frontier of the CCR model. By deciding a weight of each facet, we calculate an efficiency score.

By deciding the weight of the facets, we evaluate the efficiency score of DMU(k) as follows:

$$\bar{\theta}(k) := \sum_{j=1}^{h} w_j \bar{\theta}_j(k),$$

where $(w_1, \ldots, w_h)$ is the weight satisfying $\sum_{j=1}^{h} w_j = 1$ and $w_j \geq 0\,(j = 1, \ldots, h)$. Since $\bar{\theta}_j(k) \leq 1$, $j = 1, \ldots, h$ for each DMU(k), $0 < \bar{\theta}(k) \leq 1$.

Table 5.3: The data of eight DMUs

DMU	A	B	C	D	E	F	G	H
Input 1	2	3	3	4	5	5	6	8
Input 2	5	3	6	2	7	4	5	6
Output	1	2	3	3	2	4	3	5

We explain the approach to evaluate DMUs using a simple data in Table 5.3. In the CCR model, three facets H_1, H_2 and H_3 are used to evaluate all DMUs shown in Figure 5.1.

$$H_1 := \{(x,y) : -x_1 + y = 0\},$$
$$H_2 := \{(x,y) : -4x_1 - x_2 + 6y = 0\},$$
$$H_3 := \{(x,y) : -1.5x_2 + y = 0\}.$$

For example, C is evaluated as CCR-efficient by H_1 or H_2. However, by H_3, C is evaluated as CCR-inefficient. The efficiency scores $\bar{\theta}_i(k)$ by H_i $(i = 1, 2, 3)$ are shown in Table 5.4. In the CCR model, each DMU selects the weight which obtains the maximum efficiency score, that is, $(w_1, w_2, w_3) = (1, 0, 0), (0, 1, 0)$ or $(0, 0, 1)$. Then, $\theta^*(k) := \max\{\bar{\theta}_1(k), \bar{\theta}_2(k), \bar{\theta}_3(k)\}$ is the efficiency score of the CCR model. The facets which give the maximum efficiency score for each DMU are called *reference facet.* We calculate the efficiency score by deciding the importance of each facet.

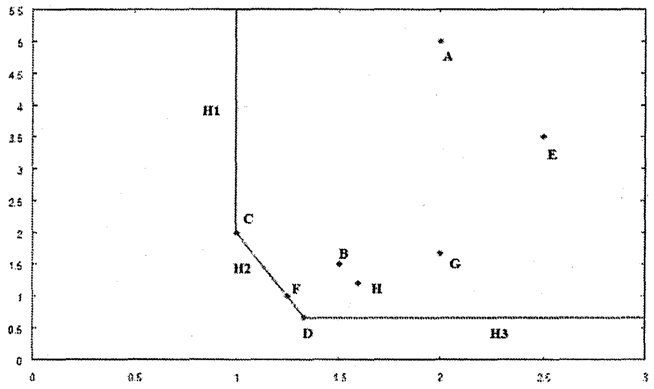


Figure 5.1: Structure of the efficient frontier

Table 5.4: Efficiency scores by each facet

DMU	$\bar{\theta}_1(k)$	$\bar{\theta}_2(k)$	$\bar{\theta}_3(k)$
A	0.5000	0.4615	0.1333
B	0.6667	0.8000	0.4444
C	1.0000	1.0000	0.3333
D	0.7500	1.0000	1.0000
E	0.4000	0.4444	0.1905
F	0.8000	1.0000	0.6666
G	0.5000	0.6207	0.4000
H	0.6250	0.7895	0.5555

First, we propose a weight based on the frequencies of the reference facets for all

DMUs. For each $i = 1, \dots, h$, let $\mathcal{H}(i) := \{j \in \{1, \dots, n\} : \bar{\theta}_i(j) = \theta^*(j)\}$. Then the weight w_i is calculated as follows:

$$w_i := \frac{|\mathcal{H}(i)|}{\sum_{j=1}^{h} |\mathcal{H}(j)|}.$$

For the data in Table 5.3, the frequency is calculated as indicated by Table 5.5. Then, we obtain $(w_1, w_2, w_3) = (0.2, 0.7, 0.1)$.

Table 5.5: The frequencies of the reference facets

DMU	Reference facets	H_1	H_2	H_3
A	H_1	1	0	0
B	H_2	0	1	0
C	H_1,H_2	1	1	0
D	H_2,H_3	0	1	1
E	H_2	0	1	0
F	H_2	0	1	0
G	H_2	0	1	0
H	H_2	0	1	0
	The frequency	2	7	1

Second, we suggest a weight based on the frequencies of the reference facets for CCR-efficient DMUs, that is, we examine the number of DMUs on each facet. For the data in Table 5.3, C, D and F are evaluated as CCR-efficient and the frequency is calculated as indicated by Table 5.6. Let $\mathcal{H}'(i) := \{j \in \{1, \dots, n\} : \bar{\theta}_i(j) = 1\}$. Then the weight w_i is calculated as follows:

$$w_i := \frac{|\mathcal{H}'(i)|}{\sum_{j=1}^{h} |\mathcal{H}'(j)|}.$$

Then, $(w_1, w_2, w_3) = (0.2, 0.6, 0.2)$. The difference of two weights is whether the decision maker adopt the ideas of the inefficient DMUs.

5.4 Example

In this section, we consider an example investigated by Wong and Beasley [42]. There are seven departments in a university listed in Table 5.7, where each department has

Table 5.6: The frequencies of the reference facets

DMU	Reference facets	H_1	H_2	H_3
A	H_1	0	0	0
B	H_2	0	0	0
C	H_1,H_2	1	1	0
D	H_2,H_3	0	1	1
E	H_2	0	0	0
F	H_2	0	1	0
G	H_2	0	0	0
H	H_2	0	0	0
The frequency		1	3	1

three inputs and three outputs. Three inputs (x_1, x_2, x_3) are the number of academic staff, academic staff salaries in thousands of pounds and support staff salaries in thousands of pounds, respectively. Three outputs (y_1, y_2, y_3) are the numbers of undergraduate students, postgraduate students and research papers, respectively.

Table 5.7: The data of seven DMUs

DMU	Inputs			Outputs			CCR efficiency score
	x_1	x_2	x_3	y_1	y_2	y_3	
A	12	400	20	60	35	17	1.0000
B	19	750	70	139	41	40	1.0000
C	42	1500	70	225	68	75	1.0000
D	15	600	100	90	12	17	0.8197
E	45	2000	250	253	145	130	1.0000
F	19	730	50	132	45	45	1.0000
G	41	2350	600	305	159	97	1.0000

In Table 5.7, six DMUs are rated as CCR-efficient and D is the only department that is rated as CCR-inefficient. In this example, there exist 51 facets forming the efficient frontier of the CCR model. In Table 5.8, we show the aggressive facet cross efficiency scores of the seven departments which coincide with the score by utilizing the optimal solutions of Problem(AGG(k)) for seven departments. In Table 5.9, we show the sum minimal facet cross efficiency scores of the seven departments which is closer cross efficiency score to Problem (AVE(k)) than the score by solving

Problem (AGG(k)). For each column in Table 5.9, we realize that the sum of the efficiency scores of all DMUs is less than or equal to the sum calculated by the aggressive facet cross efficiency method. For example, B select a weight which has 5.715 as the sum of the efficiency scores of all DMUs by solving Problem(AGG(k)). In contrast, we can obtain a weight which has 5.675 as the sum of the efficiency scores of all DMUs by using the method proposed in Section 5.2.

Table 5.8: Aggressive facet cross efficiency

DMU	Target DMU							Average	Rank
	A	B	C	D	E	F	G		
A	1.000	0.845	0.933	0.687	0.645	0.793	0.752	0.8081	2
B	0.335	1.000	0.618	1.000	0.824	0.701	0.556	0.7191	4
C	0.555	0.848	1.000	0.735	0.813	1.000	0.418	0.7669	3
D	0.069	0.755	0.280	0.820	0.367	0.236	0.206	0.3904	7
E	0.331	0.662	0.315	0.765	1.000	0.699	0.831	0.6576	5
F	0.514	1.000	0.821	0.951	1.000	1.000	0.611	0.8424	1
G	0.151	0.604	0.158	1.000	0.525	0.246	1.000	0.5264	6
Sum	2.956	5.715	4.125	5.958	5.174	4.675	4.374		

Table 5.9: Sum minimal facet cross efficiency

DMU	Target DMU							Average	Rank
	A	B	C	D	E	F	G		
A	1.000	0.757	0.793	0.687	0.490	0.793	0.752	0.7533	2
B	0.335	1.000	0.533	1.000	0.729	0.701	0.556	0.6934	4
C	0.555	0.811	1.000	0.735	0.618	1.000	0.418	0.7337	3
D	0.069	0.718	0.159	0.820	0.392	0.236	0.206	0.3713	7
E	0.331	0.705	0.485	0.765	1.000	0.699	0.831	0.6881	5
F	0.514	1.000	0.840	0.951	0.820	1.000	0.611	0.8193	1
G	0.151	0.667	0.151	1.000	0.819	0.246	1.000	0.5764	6
Sum	2.956	5.657	3.962	5.958	4.869	4.675	4.374		

In order to decide a cross efficiency score uniquely, many researchers have formulated linear or non-linear programming problems. By using the optimal solutions, a cross efficiency score is calculated. In Section 5.2, we have proposed the methods to decide a cross efficiency score uniquely by using the equations forming F_{CCR}.

Chapter 6

CONCLUSIONS

In this thesis, we have proposed four types of improvements for inefficient DMUs and five kinds of evaluation methods to rank all DMUs by analyzing all equations forming the facets of the efficient frontiers in DEA.

In Chapter 4, we have proposed four types of improvements for inefficient DMUs in the CCR model. In order to calculate flexible improvements, we have proposed three kinds of algorithms for obtaining all equations forming the efficient frontiers of the basic DEA models. By using them, we have proposed an algorithm to calculate four kinds of improvements based on each constraint condition. By this approach, we have obtained an improvement taking into account PPS of another model. Moreover, we have improved the algorithm to obtain a CCR-Pareto-efficient improvement.

In Chapter 5, we have proposed five kinds of methods to calculate the cross efficiency scores by using all equations forming F_{CCR}. In general, there exists a pair of DMUs having no dominance relationship in many of the standard DEA models. In conventional DEA models, each DMU is evaluated by using a most advantageous weight for the object DMU. By using the advantageous weights for all DMUs, the cross efficiency evaluation ranks all DMUs. In this thesis, we have regarded the coefficients of the equations forming F_{CCR} as the weights selected by DMUs. By selecting a part of facets based on the idea of the cross efficiency evaluation, we have obtained the same scores as the traditional cross efficiency evaluations without solving linear programming problems. Moreover, we have obtained a closer cross efficiency score for minimizing the sum of cross efficiency scores of the other DMUs than the score by solving the traditional formulation.

In previous DEA approaches, an improvement for inefficient DMU and ranking of all DMUs are calculated by solving linear programming problems repeatedly. In this thesis, we have proposed the approaches to obtain them by using all equations forming the facets of the efficient frontiers. We hope that this approach is widely utilized for a study of DEA in the future.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Professor Tamaki Tanaka and Associate Professor Syuuji Yamada of Niigata University for their valuable advice and help on my research. I have studied under Associate Professor Yamada during my undergraduate and master's course, and under Professor Tanaka during doctor's course at Niigata University. I wish to express my deep appreciation for their kind support and encouragement.

Also, I am grateful to Professor Eiichi Isogai of Niigata University for his comments and advice. I wish to acknowledge Professor Tetsuzo Tanino of Osaka University for his persistent and continuous encouragement.

I would like to thank Professor Nobuhiro Innami, Professor Teruo Takeuchi of Niigata University for their comments on this thesis.

Furthermore, I am grateful to all my friends and colleagues in Professor Tanaka and Associate Professor Yamada's laboratories.

Finally, I am thankful to my family for their encouragement and support.

Bibliography

[1] I. Ali and L. Seiford, *Translation Invariance in Data Envelopment Analysis*, Operations Research Letter 9 (1990) 403–405.

[2] P. Andersen and N.C. Petersen, *A Procedure for Ranking Efficient Units in Data Envelopment Analysis*, Management Science 39 (1993) 1261–1264.

[3] J. Aparicio, J.L. Ruiz and I. Sirvent, *Closest Targets and Minimum Distance to the Pareto-efficient Frontier in DEA*, Journal of Productivity Analysis 28 (2007) 209–218.

[4] R.D. Banker, A. Charnes and W.W. Cooper, *Some Models for Estimating Technical and Scale Inefficinecies in Data Envelopment Analysis*, Management Science 30 (1984) 1078–1092.

[5] R.D. Banker and R. Morey, *Efficiency Analysis for Exogenously Fixed Inputs and Outputs*, Operations Research 34(4) (1986) 513–521.

[6] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *Nonlinear Programming - Theory and Algorithms (Second Edition)*, John Wiley, New York, 1993.

[7] A. Charnes, W.W. Cooper, B. Golany, L.M. Seiford and J. Stutz, *Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions*, Journal of Econometrics 30 (1985) 91–107.

[8] A. Charnes, W.W. Cooper, A.Y. Lewin, R.C. Morey and J.J. Rousseau, *Sensitivity and Stability Analysis in DEA*, Annals of Operations Research 2 (1985) 139–156.

[9] A. Charnes, W.W. Cooper and E. Rhodes, *Measuring Efficiency of Decision Making Units*, European Journal of Operational Research 2 (1978) 429–444.

[10] A. Charnes, W.W. Cooper and E. Rhodes, *Evaluation Program and Managerial Efficiency : An Application of Data Envelopment Analysis to Program Follow Through*, Management Science 27 (1981) 668–697.

[11] A. Charnes, W.W. Cooper, L.M. Seiford and J. Stutz, *A Multiplicative Model for Efficiency Anakysis*, Socio-Economic Planning Sciences 16 (1982) 223-224.

[12] A. Charnes, W.W. Cooper, L.M. Seiford and J. Stutz, *Invariant Multiplicative Efficiency and Piecewise Cobb-Douglas Envelopments*, Operations Research Letter 2 (1983) 101–103.

[13] A. Charnes, W.W. Cooper and R.M. Thrall, *A Structure for Classifying and Characterizing Efficiency and Inefficiency in Data Envelopment Analysis*, Journal of Productivity Analysis 2 (1991) 197–237.

[14] A. Charnes, W.W. Cooper, Q.L. Wei and Z.M. Huang, *Cone Ratio Data Envelopment Analysis and Multi-objective Programming*, International Journal of System Science 20 (1989) 1099–1118.

[15] A. Charnes and L. Neralic, *Sensitivity Analysis of Additive Model in Data Envelopment Analysis*, European Journal of Operational Research 48 (1990) 332–341.

[16] W.W. Cooper, K.S. Park and J.T. Pastor, *RAM: a Range Adjusted Measure of Inefficiency for Use with Additive Models, and Relations to Other Models and Measures in DEA*, Journal of Productivity Analysis 11 (1999) 5–42.

[17] W.W. Cooper, L.M. Seiford, K. Tone, *Data Envelopment Analysis, A Comprehensive Text with Models, Applications, References and DEA-Solver Software (Second Edition)*, Springer, New York, 2006.

[18] J. Doyle and R. Green, *Efficiency and Cross-efficiency in DEA: Derivations, Meanings and Uses*, Journal of the Operations Research Society 45 (1994) 567–578.

[19] T. Entani, Y. Maeda and H. Tanaka, *Dual Models of Interval DEA and Its Extension to Interval Data*, European Journal of Operational Research 136 (2002) 32–45.

[20] R. Fare and S. Grosskopt, *A Nonparametric Cost Approach to Scale Efficiency*, Scand. J. of Economics 87 (1985) 594–604.

[21] F. Frei and P. Harker, *Projections onto Efficient Frontiers: Theoretical and Computational Extensions to DEA*, Journal of Productivity Analysis 11 (1999) 275–300.

[22] N. Hibiki, *Evaluation Techniques with a Modified Cross-efficiency in DEA*, Journal of the Operations Research Society of Japan 41(2) (1998) 229–245.

[23] N. Hibiki and T. Sueyoshi, *DEA Sensitivity Analysis by Changing a Reference Set: Regional Contribution to Japanese Industrial Development*, International Journal of Management Science 27 (1999) 139–153.

[24] G.R. Jahanshahloo, F.H. Lotfi and M. Zohrehbandian, *Finding the Piecewise Linear Frontier Production Function in Data Envelopment Analysis*, Applied Mathematics and Computation 163 (2005) 483–488.

[25] M.B. Jonathan and A.L. Adrian, *Convex Analysis and Nonlinear Optimization Theory and Examples (Second Edition)*, Springer, New York, 2006.

[26] H. Konno, P.T. Thach and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, Kluwer Academic Publishers, Dordrecht, 1997.

[27] C.A.K. Lovell and J.T. Pastor, *Units Invariant and Translation Invariant DEA Models*, Operations Research Letter 18 (1995) 147–151.

[28] S. Lozano and G. Villa, *Gradual Technical and Scale Efficiency Improvement in DEA*, Annals of Operations Research 173 (2010) 123–136.

[29] J.T. Pastor, *Translation Invariance in DEA:A Generalization* Annals of Operations Research 66 (1996) 93–102.

[30] L.M. Seiford and R.M. Thrall, *Recent Developments in DEA, The Mathematical Programming Approach to Frontier Analysis*, Journal of Econometrics 46 (1990) 7–38.

[31] J.K. Sengupta, *Data Envelopment Analysis for Efficiency Measurement in the Stochastic Case*, Computers and Operations Research 14 (1987) 117–129.

[32] T.R. Sexton, R.H. Silkman and A.J. Hogan, *Data Envelopment Analysis : Critique and Extensions. In R.H.Silkman (eds.) : Measuring Efficiency*, An Assessment of Data Envelopment Analysis (Jossey Bass, San Francisco) (1986) 73–105.

[33] M.C.A. Silva, P. Castro and E. Thanassoulis, *Finding Closest Targets in Non-oriented DEA Models: the Case of Convex and Non-convex Technologies*, Journal of Productivity Analysis 19 (2003) 251–269.

[34] A. Takeda and H. Nishino, *On Measuring the Inefficiency with the Inner-product Norm in Data Envelopment Analysis*, European Journal of Operational Research 133 (2001) 377–393.

[35] R.G. Thompson, P.S. Dharmapala and R.M. Thrall, *Importance for DEA of Zeros in Data, Multipliers, and Solutions*, The Journal of Productivity Analysis 4 (1993) 379–390.

[36] R.G. Thompson, L.N. Langemeier, C. Lee and R.M. Thrall, *The Role of Multiplier Bounds in Efficiency Analysis with Application to Kansas Farming*, Journal of Econometrics 46 (1990) 93–108.

[37] R.G. Thompson, F.D. Singleton, R.M. Thrall and B.A. Smith, *Comparative Site Evaluations for Locating a High-Energy Physics Lab in Texas*, Interfaces 16 (1986) 35–49.

[38] K. Tone and M. Tsutsui, *An Epsilon-based Measure of Efficiency in DEA - A Third Pole of Technical Efficiency*, European Journal of Operational Research 207 (2001) 1554–1563.

[39] K. Tone, *A Slack-based Measure of Efficiency in Data Envelopment Analysis*, European Journal of Operational Research 130 (2001) 498–509.

[40] Y.M. Wang and K.S. Chin, *A Neutral DEA Model for Cross-efficiency Evaluation and Its Extension*, Expert Systems with Applications 37 (2010) 3666–3675.

[41] S. Washio, S. Yamada, T. Tanaka and T. Tanino, *Improvement Indices Keeping the Feasibility in Data Envelopment Analysis*, Journal of Nonlinear and Convex Analysis 12(1) (2011) 29–48.

[42] T.H.B. Wong and J.E. Beasley, *Restricting Weight Flexibility in Data Envelopment Analysis*, Journal of the Operational Research Society 41(9) (1990) 829–835.

[43] J. Wu, L. Liang, Y. Zha and F. Yang, *Determination of Cross-efficiency under the Principle of Rank Priority in Cross-evaluation*, Expert Systems with Applications 36 (2009) 4826–4829.

[44] J. Zhu, *Super-efficiency and DEA Sensitivity Analysis*, European Journal of Operational Research 129 (2001) 443–455.