National And Kapodistrian University
Of Athens

Master Thesis

Literature Review on Data Envelopment Analysis and Applications

Author:
Panagiota Tsoutsoulopoulou

Supervisor:
Athanasia Manou

February 1, 2023

Abstract

Literature Review on Data Envelopment Analysis and Applications

by Panagiota Tsoutsoulopoulou

Data Envelopment Analysis is a group of methods designed to evaluate the productivity of organisations and businesses without the use of a priori assumptions. In this thesis we will present a bibliographic review of some of the methods used in Data Envelopment Analysis (DEA). Specifically, the CCR and BCC input and output oriented models, along with their dual, two phased forms will be analyzed in detail. We will also present certain models that deal with non-discretionary inputs, as well as categorical ones. Lastly, we will present an application of the DEA models above in different Markov queuing systems. using as input the number of servers and the capacity, and as outputs the percentage of customers that enter the system, as well as the waiting ans sojourn times.

In particular, Chapter 1 will contain simple examples that can be solved graphically. Chapter 2 will present the CCR model. Chapter 3 will present the two phases of the Dual CCR model along with both of the the output-oriented CCR models. Chapter 4 will present all the corresponding BCC models. Chapter 5 will present some alterations on the aforementioned DEA models, such as adding Categorical Variables. Finally, Chapter 6 will contain an application of the above in Markov queuing systems.

Περίληψη

Βιβλιογραφική Ανασκόπηση στην Περιβάλλουσα Ανάλυση Δεδομένων και Εφαρμογές

Η Περιβάλλουσα Ανάλυση Δεδομένων αποτελείται από ένα σύνολο μεθόδων με σκοπό την αξιολόγηση της παραγωγικότητας οργανισμών και επιχειρήσεων, χωρίς τη χρήση a priori υποθέσεων. Στην παρούσα εργασία θα παρουσιάσουμε μια βιβλιογραφική ανασκόπηση κάποιων μεθόδων που χρησιμοποιούνται στην Περιβάλλουσα Ανάλυση Δεδομένων (DEA). Συγκεκριμένα, θα αναλύσουμε ιδιαίτερα τα *input-oriented* και *output-oriented* μοντέλα CCR και BCC, μαζί με τα αντίστοιχα δυικά τους. Θα παρουσιάσουμε μοντέλα που ασχολούνται με input που δεν είναι υπό τον έλεγχο της διοίκησης, καθώς επίσης και μοντέλα που ενσωματώνουν και κατηγορικές μεταβλητές. Τέλος, θα εφαρμόσουμε τα παραπάνω μοντέλα σε διαφορετικές Μαρκοβιανές ουρές αναμονής, με input τον αριθμό των υπηρετών και την χωρητικότητα και output το ποσοστό των πελατών που μπαίνουν στο σύστημα, μαζί με τον μέσο χρόνο αναμονής και παραμονής τους στο σύστημα.

Συγκεκριμένα, το Κεφάλαιο 1 θα περιέχει απλά παραδείγματα τα οποία μπορούν να λυθούν γραφικά. Το Κεφάλαιο 2 θα εισάγει το input-oriented μοντέλο CCR. Το Κεφάλαιο 3 θα παρουσιάσει τις 2 φάσεις του δυικού του μοντέλου, καθώς και τα αντίστοιχα output-oriented μοντέλα. Το Κεφάλαιο 4 θα παρουσιάσει τα αντίστοιχα μοντέλα BCC. Το Κεφάλαιο 5 θα έχει κάποιες εναλλαγές των παραπάνω μοντέλων, προσθέτοντας, για παράδειγμα, κατηγορικές μεταβλητές. Τέλος, το Κεφάλαιο 6 θα περιέχει μια εφαρμογή αυτών στην αξιολόγηση Μαρκοβιανών ουρών αναμονής.

Acknowledgements

I would like to thank my supervisor, Dr Athanasia Manou, for her guidance and support. I'm deeply indebted to her, for all the time she dedicated to helping me with my thesis, as well as for giving me the confidence to pursue whatever I found interesting during its making. Her contribution was invaluable.

I would also like to express my gratitude to Dr Antonis Economou and Dr Apostolos Burnetas for their time, for agreeing to be members of my thesis committee, and, in general, for all the many things they taught me about Operational Research and Queuing Theory, which I used liberally in this thesis. I would, similarly, like to thank all my professors for allowing me to advance my knowledge and skills one step further.

On a more personal note, I would like to thank my parents and sisters for supporting me through this incredible experience. They, along with my boyfriend, are what kept me going.

Thank you all.

Contents

Part I

Data Envelopment Analysis

Chapter 1

An Introduction to DEA

1.1 Introduction

Data Envelopment Analysis is a way to evaluate the performance of organisations like businesses, hospitals or educational institutions, which will be referred to as *DMUs* (*Decision Making Units*). With that in mind, the measures that can be used to evaluate them are the commodities (money, work-hours, etc) used to produce the end result and the end result itself.

To measure a DMUs productivity we use the ratio

$$\frac{output}{input},$$

where output is what is produced (it can be economical gains or time saved or anything else we need to evaluate), and input is the resources needed to produce the output (man-hours, expenses, etc)[4].

When we use a single input and output to evaluate a DMU's performance, profit per worker-hours par example, the ratio is called "partial productivity measure". We will mainly consider, though, the "total factor production measures", which attempt to account for all the possible inputs and outputs so as to avoid wrongly attributing gains or not taking into account potential outputs, and thus under or over evaluating the DMU's productivity. We will use weights to each input or output used and the measure of efficiency will be

$$\frac{\text{weighted outputs}}{\text{weighted inputs}}.$$

The problem with the total factor production measures is the increased numerical complexity of using a large number of variables (inputs and outputs) and the extra difficulty of choosing the right weights.

The DEA approach does not require the user to decide on weights. Each DMU is given a favorable set of weights, i.e. the ones that maximize its efficiency score, solving a simple mathematical programming model. This way we can handle a large number of DMUs, inputs and outputs. Moreover, as it is a very well documented field already, a large amount of theory and methods have been developed and are at our disposal.

1.2 Single input - Single output

Example 1.1

The most simple example is the evaluation of a DMU using a single input and single output. Even though it does not usually reflect reality and can therefore be a subjective measure of performance (the choice of inputs and outputs affects greatly the evaluation), it will help for our purposes to first study the simplest case.

As a first example we will evaluate a set of 8 hospitals using as input the number of doctors and as output the number of cured patients that leave the hospital. The following table (Table 1.1) presents the number of doctors employed and patients cured in 8 hospitals.

Hospital	A	B	C	D	E	F	G	H
Doctors	10	5	4	6	5	3	5	9
Patients cured	5	5	2	2	3	4	4	3
Patients/Doctors	0.5	1	0.5	0.3	0.6	1.3	0.6	0.3

Table 1.1: Single Input and Single Output case

Plotting the number of Doctors on the horizontal and the number of Patients cured on the vertical axis (Figure 1.1), the slope of the line connecting each point to the origin (OA, OB, etc) corresponds to the ratio of the number of patients per number of doctors working there, i.e. it is the hospital's efficiency. The highest such slope is attained by the line from the origin through F and is called the *efficient frontier* (the red line). Hospital F, that attains the highest slope, is called *efficient*.

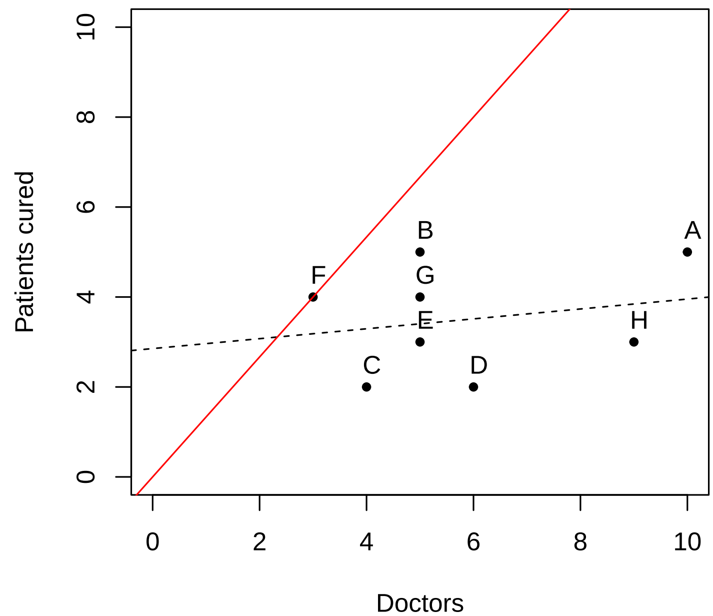


Figure 1.1: Plot of the number of the hospital's doctors to the patients cured, the frontier line (red) and the regression line (dashed)

One might be tempted to draw the regression line (the dashed line) and, as it is the average, simply consider those above it as efficient, and those below it as inefficient. But if you instead consider the efficient frontier (here OF) then you have a realistic best case scenario, and you can measure the DMUs' efficiency through their deviations from it. The hospital F, therefore, becomes a benchmark for how and where the rest of the hospitals can seek improvement. This line is called the efficient frontier. It passes through at least one of the points and all other points are encompassed -enveloped- by it, hence the name *Data Envelopment Analysis* or *DEA*, for short.

Of course, it is not really feasible that the efficient frontier stretches to infinity with a constant slope, but that will be addressed later on. Nevertheless, this line is effective for our interests. We call this *constant returns-to-scale assumption*.

When compared to F the rest of the hospitals are *inefficient* since, presumably, under the assumption that the number of Doctors is proportionate to the number of Patients cured in a week, if they had the same amount of Doctors they would cure less patients. We measure the others' efficiency (relative to F) by the ratio

$$0 \leq \frac{\text{Patients/Doctors of others}}{\text{Patients of F/Doctors of F}} \leq 1\,.$$

And we get the following table (Table 1.2)

Hospital	A	B	C	D	E	F	G	H
Relative Efficiency	0.38	0.77	0.38	0.23	0.46	1	0.46	0.23

Table 1.2: Relative Efficiency

and arrange them in order of relative efficiency.

$$1 = F > B > E = G > A = C > D = H = 0.23$$

So the less efficient ones, D and H, attain 23% of F's efficiency.

The problem now is how to make the inefficient DMUs efficient by moving them up to the efficient frontier. For C, for example, using the same methods as hospital F, it can increase the Patients they cure by $\frac{4}{3}$ (to C_2), or reduce the number of Doctors by $\frac{5}{2}$ (to C_1) (Figure 1.2).

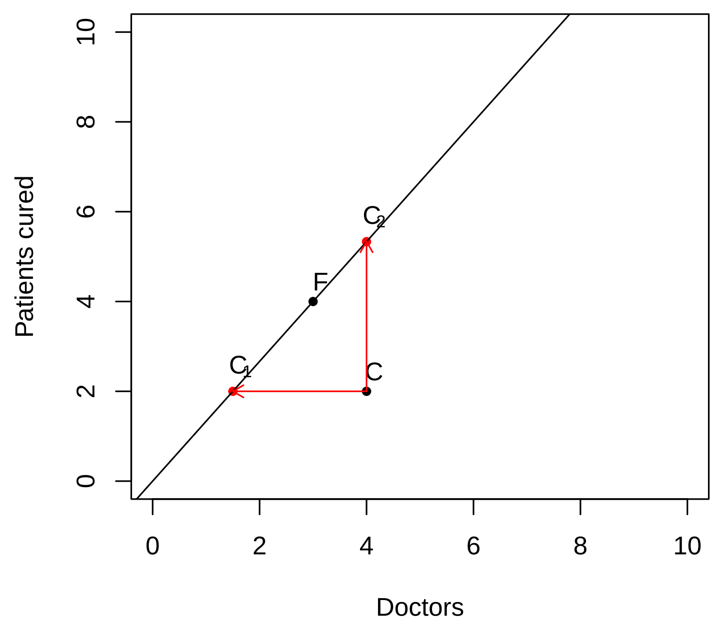


Figure 1.2: Improvement of C

Using the ratio of ratios (the relative efficiency), the values are now independent of the units of measure, which is an important property in science and engineering.

Unfortunately, this formula can only be used in the simplest case of one input and one output and any attempt to extend it to use in the cases of multiple inputs or outputs encounters many difficulties.

1.3 Two inputs - Single output

Example 1.2

To extend the method to multiple inputs and outputs and see how they could be handled we use a case with two inputs and one output. Using the same group of hospitals for example, and examining the number of doctors and the number of patients cured within a week, but adding also the number of Intensive Care Units (ICUs) in the hospital, we can study more accurately the efficiency of the hospitals. (Table 1.3)

Hospital	A	B	C	D	E	F	G	H
ICUs	6	9	13	10	7	11	6	8
Doctors	10	5	4	6	5	3	5	9
Patients cured	5	5	2	2	3	4	4	3

Table 1.3: Two Inputs and One Output Case

To study this more easily we unitize the outputs, under the constant returns-to-scale assumption. As a result we get Table 1.4, where:

$$\text{input1} = \frac{\text{ICUs}}{\text{patients cured}}$$

and

$$\text{input2} = \frac{\text{doctors}}{\text{patients cured}} .$$

Hospital	A	B	C	D	E	F	G	H
(input1)	1.2	1.8	6.5	5	2.33	2.75	1.5	2.67
(input2)	2	1	2	3	1.67	0.75	1.25	3
(output)	1	1	1	1	1	1	1	1

Table 1.4: Unitized Output, under constant returns-to-scale assumption

So the new inputs correspond to the ones needed to produce one unit of output. The most efficient, then, are the ones that use less inputs, and the procedure is greatly simplified.

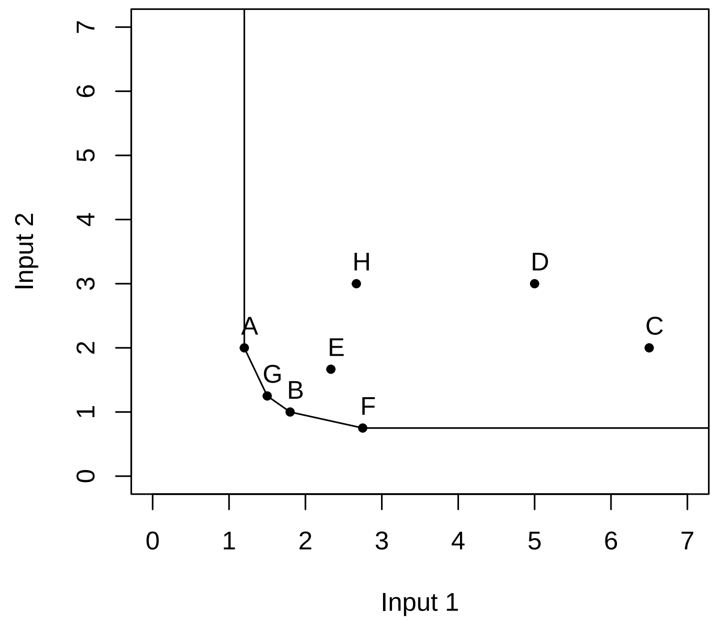


Figure 1.3: Plot of doctors to ICUs per patient cured

Plotting it (Figure 1.3), we identify the line connecting A, G, B and F as the efficient frontier, because, according to empirical evidence, no unit on it can improve its efficiency by decreasing its input values without worsening the other (F is still considered efficient). We can envelop all the data points within the region enclosed by the efficient frontier, the horizontal line passing through F, and the vertical line through A. The region enclosed is called the *production possibility set*, and includes all the production rates we assume, through solely empirical evidence, to be possible.

The (relative) efficiency, then, of a unit (E, for example) can be measured using the ratio:

$$\frac{OP}{OE} = 0,6923$$

where P is the point where OE intersects the efficient frontier (Figure 1.4).

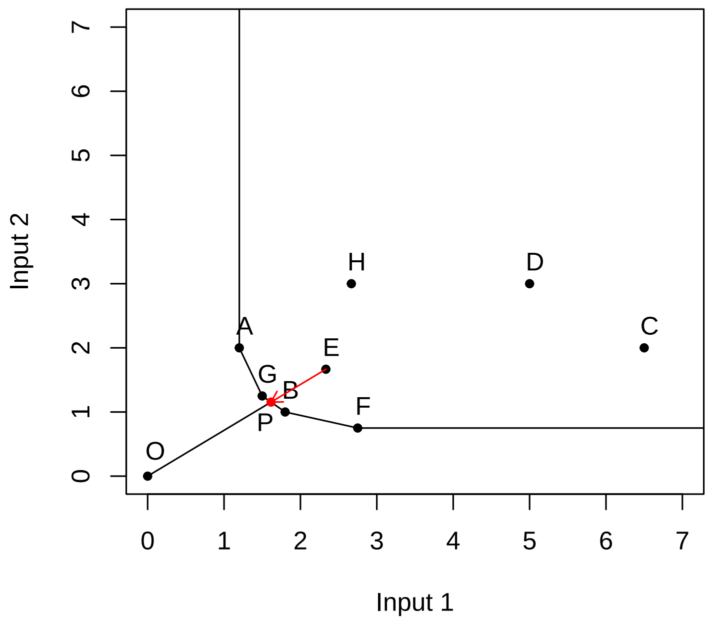


Figure 1.4: Improvement of Hospital E

Since point P is in the line between B and G, E's inefficiency can be judged and computed through a combination of B and G. B and G are then referred to as E's *reference set*. Of course, the reference set is different for each DMU. For example, H's reference set is $\{A, G\}$. Also, since most points' reference set includes B, B is referred to as a *representative Unit*.

So E's efficiency can be improved by implementing a mix of the methods that are used by the hospitals G and B, thereby reducing input 1 by 0.72 and input 2 by 0.51, i.e., without changing the inputs' proportions. However, it can also simply be improved by reducing input 1 by 0.83 and input 2 by 0.42, and thus moving it to G (by implementing the same methods as G), or moving it to any other point in the frontier line.

1.4 Single input - Two outputs

Example 1.3:

Using Example 1.1, again, we add a new output; the net profit produced by each hospital in a week (in thousands of dollars). So, using the number of doctors as the single input and the number of patients cured in a week as the other output we get Table 1.5.

Hospital	A	B	C	D	E	F	G	H
Doctors	10	5	4	6	5	3	5	9
Patients cured	5	5	2	2	3	4	4	3
Net	36	15	9	2	1	5	5	6

Table 1.5: One Input and Two Outputs Case

This time we unitize the inputs under the assumption of constant returns-to-scale, so for simplicity's sake we present the unitised table (Table 1.6), where output1 can be perceived as the number of patients cured per doctor and output2 as the net produced by each doctor.

Hospital	A	B	C	D	E	F	G	H
(input)	1	1	1	1	1	1	1	1
(output1)	0.5	1	0.5	0.33	0.6	1.33	0.8	0.33
(output2)	3.6	3	2.25	0.33	0.2	1.67	1	0.67

Table 1.6: Unitized Input, under constant returns-to-scale assumption

As we can see in Figure 1.5, the efficient frontier this time consists of the line connecting A, B and F, because this time the hospitals which produce more output using the same amount of input (1 doctor) are naturally more efficient. This means that the hospitals C, D, E, G and H are inefficient and their efficiency can be measured by projecting them to the efficient frontier.

The production possibility set in this case is enclosed between the efficient frontier and the axes, and a hospital's efficiency can be measured and improved by moving its coordinates (the hospital's outputs) towards the efficient frontier .

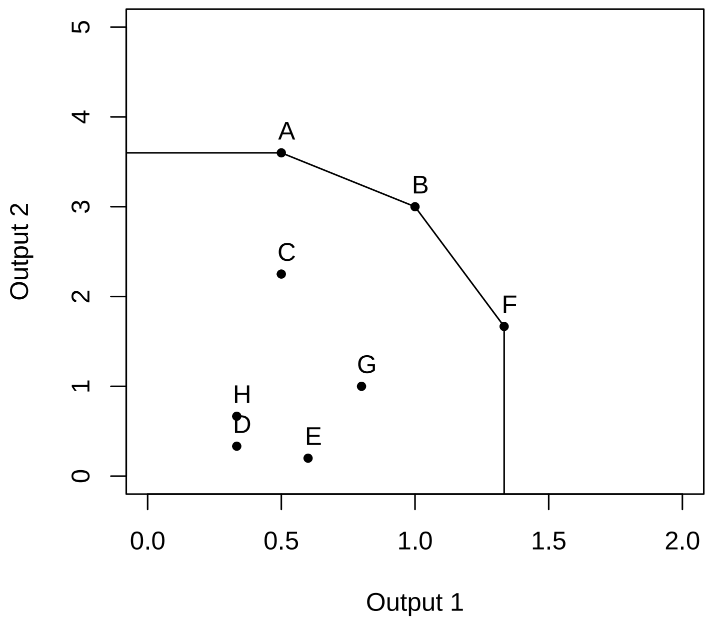


Figure 1.5: Plot of Output2 to Output1

For example, the efficiency of hospital H can be measured by the ratio

$$\frac{OH}{OP} = 0.2857,$$

where P is the point of intersection between OH and the frontier line (Figure 1.6).

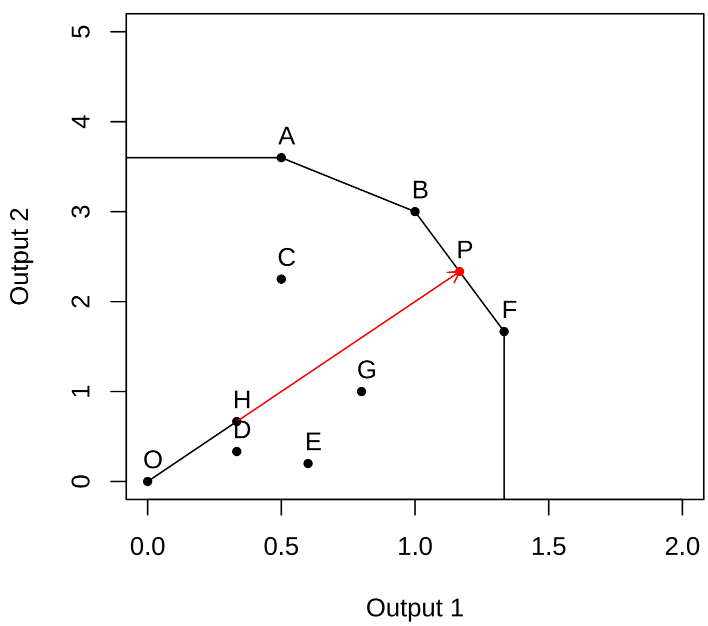


Figure 1.6: Improvement of Hospital H (purely technical inefficiency)

This ratio is referred to as a radial measure. To evaluate the distance, we use the Euclidean distance, though any other measure can also be used. Of course since the ratio is calculated over the probability set's distance to O, the ratio will always be between 0 and 1. It represents the proportion of the achieved outputs to the potential outputs that could be achieved if it were implementing a mixture of the measures used by B and F, without changing the proportions of the outputs produced. This kind of inefficiency is referred to as *ratio inefficiency*, whereas the inefficiencies caused by only some (not all) outputs (or inputs) is then referred to as *mix (technical) inefficiency*.

If, for example, there was another hospital (hospital I) that was on the vertical line that passes through A (Figure 1.7), then I's inefficiency is a mixed inefficiency since its efficiency can be improved by increasing only the number of patients cured, while keeping the Net fixed. Similarly, we define purely technical inefficiency and mix inefficiency in the case of 2 inputs and 1 output.

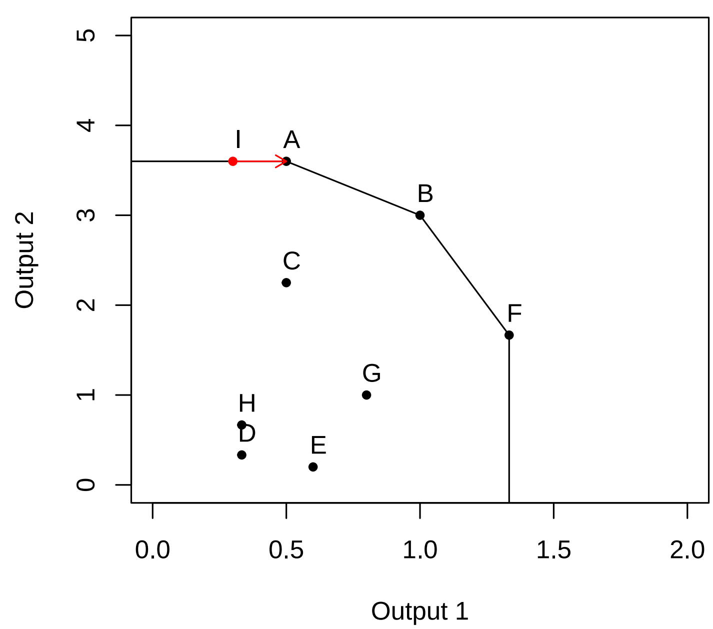


Figure 1.7: Improvement of Hospital I (mix technical inefficiency)

1.5 Fixed and Variable Weights

Example 1.4

Thus far, the examples used have at most two inputs or outputs and can therefore be easily solved graphically by unitizing the single input or output. The problem that presents itself with multiple values is the complexity of the problem. It is necessary to find a solution that is not subjective or too complex.

Say, for example, in the hospital example, if we were to use all four of the inputs and outputs (Table 1.7).

Hospital	A	B	C	D	E	F	G	H
Doctors	10	5	4	6	5	3	5	9
ICUs	6	9	13	10	7	11	6	8
Patients cured	5	5	2	2	3	4	4	3
Net	36	15	9	2	1	5	5	6

Table 1.7: Two Inputs and Two Outputs Case

One way to measure the efficiency is to assign weights to each input and output. If we choose the weights arbitrarily those are called *Fixed Weights*. For example, we use the weights such, that

$$v_1(\text{weight for the doctors}) : v_2(\text{weight for the ICUs}) = 1 : 6$$

$$u_1(\text{weight for the patients cured}) : u_2(\text{weight for the net}) = 2 : 1$$

We get the "virtual" input and output as presented on the Table 1.8, by adding the weighted inputs and outputs respectively.

Hospital	A	B	C	D	E	F	G	H
virtual input	46	59	82	66	47	69	41	57
virtual output	46	25	13	6	7	13	13	12
fixed ratio	1	0.42	0.16	0.09	0.15	0.19	0.32	0.21

Table 1.8: Virtual Input and Output Using Fixed Weights

You will find that the weights were chosen so as to normalize the ratio, with a maximum of unity. But still, they were chosen arbitrarily, and so it's not easy to justify the choices or draw any objective conclusion about the efficiency of the hospitals, since the amount of influence the weights have on each hospitals efficiency is unclear.

Any conclusion we infer using fixed weights will therefore be subjective to the set of weights we choose. To avoid this, DEA uses *Variable Weights* which are deduced directly form the data, thus avoiding making a priori assumptions by choosing them manually. The weights are chosen in a manner that assigns the best set of weights to each DMU, maximizing its relative efficiency, while normalizing the ratio, so as to have a maximum of one in all hospitals with each choice of weights (different, now in every DMU). We use one of the DEA methods, CCR (Charnes-Cooper-Rhodes, 1978)[3], to compute the best set of weights by assuming constant returns-to-scale, we get the following results (Table 1.9). The method will be presented and analysed in the following chapters, where we will see that it can also be used to identify the DMU's reference set, and the DMU's inefficiency when compared to it.

Hospital	A	B	C	D	E	F	G	H
Fixed	1	0.42	0.16	0.09	0.15	0.19	0.32	0.21
CCR	1	1	0.68	0.35	0.69	1	1	0.51

Table 1.9: Relative Efficiency using Fixed VS CCR Weights

We see that, while with the fixed weights A is the only efficient one, using the CCR method B, F and G are also perceived as efficient. The number of efficient DMUs is increased, likely because each Unit is judged using weights more beneficial to it (they maximize their relative efficiency).

1.6 Conclusion

In this chapter we have outlined some methods to measure a DMU's efficiency graphically, in the simple cases of one or two inputs and outputs by using a simple ratio of output to input. In the next chapters we will introduce and analyze methods to expand the idea to multiple outputs and inputs, using a generalized ratio.

A simple example of that was presented above using fixed weights, applied equally to all DMUs, but as they produce a very subjective measure of efficiency, we will use mathematical programming to give weights to the outputs and inputs of each DMU. And, in the interest of being fair judges of efficiency, the weights will be computed in a way that maximizes each DMUs relative efficiency.

An additional advantage of using the DEA approach of variable weights is that it can be used to recognize a DMU's reference set, and measure the inefficiency of its methods when compared to the methods used by the DMUs in the reference set, as well as its source.

All the DMUs examined each time, were assumed to use the same kind of inputs to produce the same kind of outputs, and all the items and weights were assumed to be positive. This assumption will remain in the first chapters, and then it will be relaxed.

Chapter 2

The Basic CCR model

2.1 Introduction

We assume that we have K DMUs with n inputs and m outputs each. To measure a DMU's efficiency in the case of multiple inputs and outputs we create the ratio of a single virtual input and a virtual output by assigning weights to the different items and adding them up. Specifically, the virtual input and output of a DMU o are

$$\text{virtual input} = v_1 x_{1o} + \ldots + v_n x_{no}\,, \tag{2.1}$$

$$\text{virtual output} = u_1 y_{1o} + \ldots + u_m y_{mo}\,. \tag{2.2}$$

The ratio then becomes:

$$\theta = \frac{\text{virtual output}}{\text{virtual input}} = \frac{u_1 y_{1o} + \ldots + u_m y_{mo}}{v_1 x_{1o} + \ldots + v_n x_{no}}\,, \tag{2.3}$$

where our goal becomes to maximize it over the variable weights.

The optimal weights will vary between DMUs. They are derived from the data instead of being fixed in advance, so that each DMU is assigned the best possible set of weights for it and is thus judged in the best possible light.

To secure relative comparisons the evaluation is done in relation to a group of other DMUs of **similar structure and size**, but each with a degree of managerial freedom.

2.2 The CCR model

We follow the properties below to determine the appropriate set of inputs and outputs.

Selection of Inputs and Outputs

1. Positive numerical data are available for every input and output.
2. The inputs and outputs should affect the DMUs' efficiency.
3. In principle smaller inputs and larger outputs are to be preferable for the DMU's efficiency.
4. The measurement units don't need to be congruent.

Now, let n input items and m output items be selected adhering to the properties 1-4. We will attempt to model the optimization problem.

Modeling

Given the data, we treat one DMU_i at a time, so there are K optimization problems to solve. In conclusion, we have:

- K DMUs
  - $\text{DMU}_1, \ldots, \text{DMU}_K$
- n input items
  - $\text{DMU}_k : \underline{x_k} = (x_{1k}, \ldots, x_{nk})$
- m output items
  - $DMU_k : \underline{y_k} = (y_{1k}, \ldots, y_{mk})$
- variable weights
  - $v = (v_1, \ldots, v_n)$, $v_i \geq 0$, $i = 1, \ldots, n$, the weights for the input items
  - $u = (u_1, \ldots, u_m)$, $u_j \geq 0$, $j = 1, \ldots, m$, the weights for the output items

The data can also be presented in matrix form simply as $X = (x_{ik})$ and $Y = (y_{jk})$. Let the DMU that is to be evaluated each time be designated as DMU_o, where $o = 1, \ldots, K$.

The maximization problem thus becomes a fractional programming problem. If we leave it at that, then the value $\theta^\star$ is unbounded ($\theta^\star = \infty$), so we use normalizing constraints to the ratio for each DMU:

$$\frac{\sum_{j=1}^{m} u_j y_{jk}}{\sum_{i=1}^{n} v_i x_{ik}} \leq 1, k = 1, \ldots, K. \tag{2.4}$$

Also, we make the assumption that at least one input and output are positive and have positive weights, but for simplicity's sake for now we leave that out.

The problem for DMU_o becomes:

$$(FP) \qquad \max \theta = \frac{\sum_{j=1}^{m} u_j y_{jo}}{\sum_{i=1}^{n} v_i x_{io}} \tag{2.5}$$

$$s.t. \qquad \frac{\sum_{j=1}^{m} u_j y_{jk}}{\sum_{i=1}^{n} v_i x_{ik}} \leq 1, \qquad k = 1, \ldots, K \tag{2.6}$$

$$v_1, \ldots, v_n \geq 0 \tag{2.7}$$

$$u_1, \ldots, u_m \geq 0\,. \tag{2.8}$$

The objective here is to find the weights v_i and u_j that maximize the ratio of virtual output to virtual input, while the weighted ratio (using these weights) for each DMU is at most one. If we remove that constraint then the optimum ratio is, obviously, infinite. But since that does not give us any information about the DMU's actual efficiency, or its reference set, we bound all the ratios to one (it could be any number, as long as it is common for all DMUs, 100, for example), so that the DMUs that reach the bound can be considered to have the best possible outcome and the optimal solution can be seen as the inefficiency (or efficiency if $\theta^\star = 1$) of the DMU in relation to the best possible scenarios.

Of course, for the fractions to be properly defined, the weights cannot all be zero at the same time. Seeing it in managerial terms, we simply assume that all inputs and outputs have some none zero value, and thus the weights which define that value in the ratio are positive.

It can be proven that this factional problem can be equivalently transformed into a linear problem, which is much simpler, by setting the denominator of θ to one, so we transform the problem above to the equivalent linear one.

Then the problem becomes linear with the new objective faction

$$\theta = \sum_{j=1}^{m} u_j y_{jo}\,,$$

variables v_i and u_j, and the added constraint

$$\sum_{i=1}^{n} v_i x_{io} = 1\,.$$

So the equivalent linear problem, which we solve for each DMU is:

$$
\begin{aligned}
(LP) \qquad \max\theta &= \sum_{j=1}^{m} u_j y_{jo} && (2.9)\\
s.t.\quad \frac{\sum_{j=1}^{m} u_j y_{jk}}{\sum_{i=1}^{n} v_i x_{ik}} &\le 1, \qquad k = 1,\ldots,K && (2.10)\\
\sum_{i=1}^{n} v_i x_{io} &= 1 && (2.11)\\
v_1,\ldots,v_n &\ge 0 && (2.12)\\
u_1,\ldots,u_m &\ge 0\,. && (2.13)
\end{aligned}
$$

In matrix form, it can be written as follows.

Let

$$A = \begin{pmatrix} y_{11} & \ldots & y_{m1} & -x_{11} & \ldots & -x_{n1} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ y_{1K} & \ldots & y_{mK} & -x_{1K} & \ldots & -x_{nK} \\ 0 & \ldots & 0 & x_{1o} & \ldots & x_{no} \\ 0 & \ldots & 0 & -x_{1o} & \ldots & -x_{no} \end{pmatrix} \in \mathbb{R}^{(n+2)\times(n+m)},\ \mathbf{b} = \begin{pmatrix} 0 \\ \ldots \\ 0 \\ 1 \\ -1 \end{pmatrix} \in \mathbb{R}^{n+2}\,,$$

and $\mathbf{c} = (y_{1o},\ldots,y_{mo},0,\ldots,0)' \in \mathbb{R}^{n+m}$.

Then, if $\mathbf{z} = (u_1,\ldots,u_m,v_1,\ldots,v_n)' \in \mathbb{R}^{n+m}$ is the variable, the problem takes the form

$$
\begin{aligned}
\max\theta &= \mathbf{c}'\mathbf{z}\\
s.t.\quad A\mathbf{z} &\le \mathbf{b}\\
\mathbf{z} &\ge \mathbf{0}\,,
\end{aligned}
$$

and it becomes a simple linear problem that can be solved either manually or using well known algorithms, like Simplex. More easily, it can be solve by solving its dual problem. It will be presented in the following chapter.

Supposing that the optimal solution to the linear problem is $(\theta^\star, v^\star, u^\star)$ for DMU$_o$, then we define the CCR-efficiency as follows.

Definition 2.1 CCR-efficiency

1. If $\theta^\star = 1$, and there exists at least one optimal $(v^\star, u^\star)$ with $v^\star > 0$ and $u^\star > 0$, then DMU$_o$ is *CCR-efficient*.

2. Otherwise, DMU$_o$ is *CCR-inefficient*

So a DMU is CCR-inefficient when (1) $\theta^\star < 1$, or (2) at least one element of $(v^\star, u^\star)$ is zero for every solution of the linear problem. The second case will be better explained in the next Chapter using the dual linear problem.

In the first case, when $\theta^\star < 1$ there must necessarily exist at least one DMU which satisfies the equality in the inequality (2.10), since otherwise the solution can be (and thus must be) greater than $\theta^\star$.

Let the set of such Units be:

$$\mathbf{E_o}' = \left\{ k : \sum_{j=1}^{m} u_j^\star y_{jk} = \sum_{i=1}^{n} v_i^\star x_{ik} \right\}.$$

The subset $\mathbf{E_o}$ of $\mathbf{E_o}'$ composed of CCR-efficient DMUs is referred to as DMU$_o$'s *reference set*, or its *peer group*. It is because those DMUs exist that we are forced to categorize DMU$_o$ as inefficient. The set spanned by it makes up the *efficient frontier of DMU$_o$*.

2.3 Meaning of the Optimal Weights

The $(v^\star, u^\star)$ obtained by solving the linear problem is a set of optimal weights for DMU$_o$, and using them the ratio is evaluated by

$$\theta^\star = \sum_{j=1}^{m} u_j^\star y_{jo} \,.$$

They are the most favorable weights for DMU$_o$ and each weight is optimal for the corresponding item. Its magnitude expresses how highly the item is evaluated; what worth its contribution is given when evaluating the DMU, while the product $u_j^\star y_{jo}$ or $v_i^\star x_{io}$ can be construed as the item's total contribution. So the weights show not only which items contribute to the DMU's efficiency, but also the extend at which they do.

2.4 Explanatory Examples

We will illustrate the methods using the example used in Chapter 1. So we have a set of 8 hospitals.

Example 2.1

For the first example, we will use as input the doctors working there and as output the number of patients cured in a week. (Table 2.1)

Hospital	A	B	C	D	E	F	G	H
Doctors	10	5	4	6	5	3	5	9
Patients cured	5	5	2	2	3	4	4	3

Table 2.1: Single Input and Single Output case

To measure the efficiency of hospital A, for instance, we create the following linear problem:

$$
\begin{aligned}
< A > \ \max \theta &= 5u \\
s.t. \quad 10v &= 1 \\
5u \leq 10v \ (A), &\qquad 3u \leq 5v \quad (E) \\
5u \leq 5v \quad (B), &\qquad 4u \leq 3v \quad (F) \\
2u \leq 4v \quad (C), &\qquad 4u \leq 5v \quad (G) \\
2u \leq 6v \quad (D), &\qquad 3u \leq 9v \quad (H) \\
v, u \geq 0. &
\end{aligned}
$$

The optimal solution is given by $(v^\star = 0.1, u^\star = 0.075)$. And so the CCR-efficiency of hospital A is $\theta^\star = 0.375$. Its reference set is $E_A = \{F\}$, as can be seen by substituting the optimal weights in each of the constraints. That means essentially that the performance of F is what forces A to be inefficient even when using the best possible weights for A.

Similarly, the efficiency of hospital F, or any other hospital, can be evaluated using the same inequality constraints. Specifically, for hospital F we have the following problem.

$$
\begin{aligned}
< F > \ \max \theta &= 4u \\
s.t. \quad & 3v = 1 \\
& 5u \leq 10v \ (A), \qquad 3u \leq 5v \quad (E) \\
& 5u \leq 5v \quad (B), \qquad 4u \leq 3v \quad (F) \\
& 2u \leq 4v \quad (C), \qquad 4u \leq 5v \quad (G) \\
& 2u \leq 6v \quad (D), \qquad 3u \leq 9v \quad (H) \\
& v, u \geq 0.
\end{aligned}
$$

The optimal solution here is obtained using the weights ($v^\star = 0.333, u^\star = 0.25$), and F's efficiency is $\theta^\star = 1$. So, by definition, since $\theta^\star = 1$ and $v^\star, u^\star > 0$, F is CCR-efficient. We can also easily see that its reference set is consisted only of itself.

Proceeding in the same manner with the rest of the hospitals, we find that F is the only efficient one. In the following table (Table 2.2) we present all the hospitals' efficiency, along with their peer group.

DMU	CCR ($\theta^\star$)	Peer Group
A	0.375	F
B	0.75	F
C	0.375	F
D	0.25	F
E	0.45	F
F	1	F
G	0.6	F
H	0.25	F

Table 2.2: The Hospitals' Efficiency and Peer Group

Those results track with the results we got in the first chapter by using the simple output-to-input ratio score, in regards to whether a hospital is efficient.

Example 2.2

We expanding the premises of this example to two inputs and a single output as in **Example 1.2**, by adding the number of ICUs as an extra input (Table 2.3).

The problem we solve to evaluate A's efficiency here is the following:

Hospital	A	B	C	D	E	F	G	H
Doctors	10	5	4	6	5	3	5	9
ICUs	6	9	13	10	7	11	6	8
Patients cured	5	5	2	2	3	4	4	3

Table 2.3: Two Inputs and One Output Case

$$
\begin{aligned}
< A > \ \max \theta &= 5u \\
s.t. \quad & 10v_1 + 6v_2 = 1 \\
& 5u - 10v_1 - 6v_2 \leq 0 \ (A), \qquad 3u - 5v_1 - 7v_2 \leq 0 \ (E) \\
& 5u - 5v_1 - 9v_2 \leq 0 \ (B), \qquad 4u - 3v_1 - 11v_2 \leq 0 \ (F) \\
& 2u - 4v_1 - 13v_2 \leq 0 \ (C), \qquad 4u - 5v_1 - 6v_2 \leq 0 \ (G) \\
& 2u - 6v_1 - 10v_2 \leq 0 \ (D), \qquad 3u - 9v_1 - 8v_2 \leq 0 \ (H) \\
& v_1, v_2, u \geq 0.
\end{aligned}
$$

This problem can easily be solved by a linear programming code. The optimal solution is $\theta^\star = 1$ and it's achieved by the weights $(v_1^\star = 0.04, v_2^\star = 0.1, u^\star = 0.2)$, so it is CCR-efficient. Its reference set is $E_A = \{A, G\}$.

Doing the same for the rest of the hospitals we find that (Table 2.4):

DMU	CCR ($\theta^\star$)	Peer Group
A	1	A, G
B	1	B, G
C	0.39	B, F
D	0.34	B, G
E	0.69	B, G
F	1	B, F
G	1	A, G
H	0.51	A, G

Table 2.4: The Hospitals' Efficiency and Peer Group

Example 2.3

Now we move on to a new example. Say you have 7 universities, and you wish to evaluate their efficiency using the number of papers they produce in a week as an output, when compared to the number of professors they employ, as well as the amount of money they allot for projects. The data is given in the following table (Table 2.5).

Universities	A	B	C	D	E	F	G
Professors	4	6	10	4	6	3	6
Money allotted	4	8	11	7	3	9	6
Papers	1	2	3	2	2	1	3

Table 2.5: Single Output and Two Inputs Case

To evaluate the efficiency of DMU D, we have the following LP.

$$
\begin{aligned}
< D > \ & \max \theta = 2u \\
s.t. \quad & 4v_1 + 7v_2 = 1 \\
& u - 4v_1 - 4v_2 \leq 0 \quad (A), \qquad 2u - 6v_1 - 3v_2 \leq 0 \quad (E) \\
& 2u - 6v_1 - 8v_2 \leq 0 \quad (B), \qquad u - 3v_1 - 9v_2 \leq 0 \quad (F) \\
& 3u - 10v_1 - 11v_2 \leq 0 \quad (C), \qquad 3u - 6v_1 - 6v_2 \leq 0 \quad (G) \\
& 2u - 4v_1 - 7v_2 \leq 0 \quad (D) \\
& u, v_1, v_2 \geq 0.
\end{aligned}
$$

The solution to this problem is $\theta^\star = 1$ and is achieved using the weights $u^\star = 0.5$ and $v^\star = (0.25, 0)$. So D looks, at first glance, to be efficient, but one of the optimal weights, $v_2^\star$ is zero. To ascertain whether D is really efficient, we assign a small positive value to v_2, $\varepsilon > 0$, so that $4v_1 + 7\varepsilon = 1$, meaning $v_1 = 0.25 - 1.75\varepsilon$. Replacing v_1 with its equivalent in the constraint forced by G, we get:

$$u \leq 0.5 - 1.5\varepsilon.$$

Meaning $\theta = 2u \leq 1 - 3\varepsilon < 1$, since $\varepsilon > 0$. So university D is not really CCR-efficient when the money spent on projects is taken into account.

The inefficiency of D can also be observed when compared to G as, keeping with the constant returns-to-scale assumption, G can produce 1 paper a week, using less money than D, with the same number of professors. This deficiency is concealed in the solution, because the optimal weight for the money allotted is zero, and thus its contribution to D's inefficiency is diminished. As it is a single input, two output case, it can also be presented graphically (Figure 2.1).

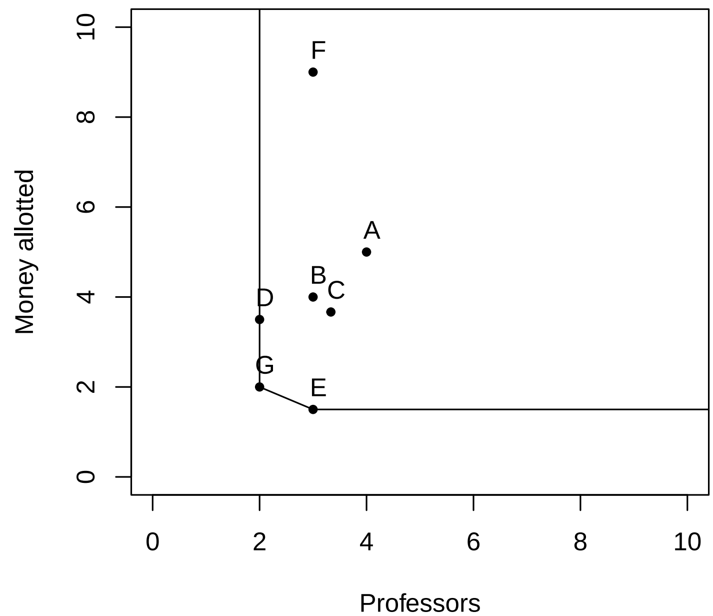


Figure 2.1: Plot of Professors-per-Paper to Money-per-Paper

In most cases, it is not as easy to identify the mix inefficiencies using only this model. They will be identified much simpler in the next chapter by solving the dual problem, that minimizes those kind of shortfalls. There, any non-zero solution indicates mix technical inefficiency.

A DMU such as D that has a ratio $\theta^\star = 1$, but presents a shortage of inputs, or an excess of outputs, is referred to as *ratio efficient but mix inefficient*.

The following table (Table 2.6) shows the CCR-efficiency of all the Universities, along with their peer group and their optimal weights.

DMU	CCR ($\theta^\star$)	Peer Group	u	v_1	v_2
A	0.5	D, G	0.5	0.25	0
B	0.67	D, G	0.33	0.17	0
C	0.6	D, G	0.2	0.1	0
D	1	D, G	0.5	0.25	0
E	1	E, G	0.5	0.08	0.17
F	0.67	D, G	0.67	0.33	0
G	1	D, G	0.33	0.17	0

Table 2.6: The Universities' Efficiency and Peer Group

As we can see, even the CCR-efficient DMUs have more than themselves in their peer group, when it would make sense to have only themselves.

2.5 Conclusion

In this chapter we presented the CCR mathematical model, which is the most basic DEA model. Using it, we can more accurately, and more analytically, assign each DMU with an efficiency score, and spot pure technical inefficiencies, as well as include more than two items on which to base their evaluation.

For each DMU, we formed the variables virtual input and virtual output, and tried to maximize the latter, while keeping the former fixed. The weights used to form the virtual input and output, tend to vary from DMU to DMU, but each set of wights has to form a positive virtual-input-to-virtual-output ratio, that is less than unity when applied to all other DMUs. Henceforth, those weights will be called *modifiers* to distinguish them from other commonly used meanings of the word, as they differ from them in that they are not fixed for all DMUs.

We also properly defined CCR-efficiency, and efficient frontiers for inefficient DMUs.

Chapter 3

The CCR Model and Production Correspondence

3.1 Introduction

In this chapter we will introduce and properly define the *production possibility set* in the CCR model. Then, we will present the dual problem which is a linear problem that when solved can give us the shortfalls and excesses that cause mixed inefficiencies. CCR-efficiency will then be redefined using those input excesses and output shortfalls.

There are two versions of CCR-models. The *input-oriented model*, which minimizes the virtual input while ensuring at least the given amount of output, and the *output-oriented model*, which maximizes the virtual output while keeping the input under the observed values.

3.2 Production Possibility Set

In this chapter the assumption that all input and output vectors must be strictly positive is relaxed. We assume all data to be semi-positive, which means that as long as at least one component of each vector is positive, the rest can be simply non-negative. We call a pair of semi-positive input and output vectors, $(\underline{x}, \underline{y})$ with $\underline{x} \in \mathbb{R}^n, \underline{y} \in \mathbb{R}^m$, an *activity*. The components of each activity can be seen as a semi-positive orthant point in a $(n+m)$-dimensional linear vector space, in which the superscript n and m express the number of dimensions required to express inputs and outputs respectively.

The set of feasible activities is called the *production possibility set* and is denoted by P.

Properties of the Production Possibility Set, P

1. All the observed activities belong to P
2. If any activity $(\underline{x}, \underline{y})$ belongs to P, the activity $(t\underline{x}, t\underline{y})$ belongs to P for any positive scalar t. We call this property *constant returns-to-scale assumption.*
3. For any activity $(\underline{x}, \underline{y})$ in P, any semi-positive activity $(\underline{x_0}, \underline{y_0})$ with $\underline{x_0} \geq \underline{x}$ and $\underline{y_0} \leq \underline{y}$ is also in P. That is, any activity with input no less than $\underline{x}$ and output no more than $\underline{y}$ is feasible.
4. Any semi-positive linear combination of activities in P belongs to P.

In matrix form, we can define the production possibility set satisfying the properties above as

$$P = \{(\underline{x}, \underline{y}) \mid \underline{x} \geq X\underline{\lambda},\ \underline{y} \leq Y\underline{\lambda},\ \underline{\lambda} \geq \underline{0}\},$$

where $\underline{\lambda}$ is a semi-positive vector in $\mathbb{R}^K$, $X = (x_{ik})$, $Y = (y_{jk})$.

In the example in Chapter 1, for instance, the production possibility set is defined by the line passing through F, and consists of all points that are between it and the axis of Doctors, as seen in Figure 3.1.

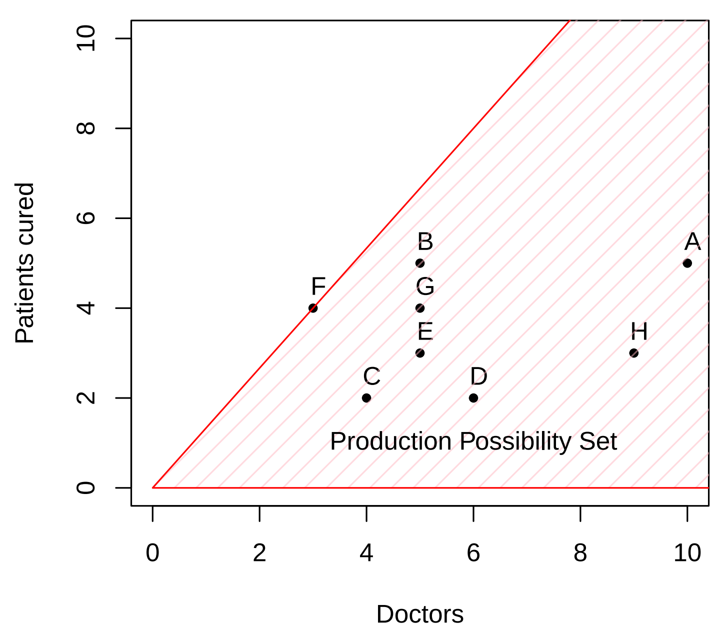


Figure 3.1: Production Possibility Set

3.3 The CCR Dual Problem

We present again the matrix form of the LP problem, where the row vectors $\underline{v}$ and $\underline{u}$, the input and output multipliers respectively, are used as variables for the maximization of the virtual scale.

$$
\begin{aligned}
(LP_o) \quad & \max\ \underline{u}'\underline{y_o} \\
& s.t. \qquad \underline{v}'\underline{x_o} = 1 \\
& \qquad \underline{u}Y - \underline{v}X \le \underline{0} \\
& \qquad \underline{u} \ge \underline{0},\ \ \underline{v} \ge \underline{0}\,,
\end{aligned}
$$

or equivalently,

$$
\begin{aligned}
(LP) \quad & \max \theta = \sum_{j=1}^{m} u_j y_{jo} \\
s.t. \quad & \sum_{i=1}^{n} v_i x_{io} = 1 \\
& \sum_{j=1}^{m} u_j y_{jk} - \sum_{i=1}^{n} v_i x_{ik} \le 0, \qquad k = 1, \ldots, K \\
& v_1, \ldots, v_n \ge 0 \\
& u_1, \ldots, u_m \ge 0\,.
\end{aligned}
$$

This problem has $K+1$ constraints and $m+n$ variables. The dual would then have $m+n$ constraints and $K+1$ variables. In linear problems, it is generally easier, less complex, to solve a problem with fewer constraints, and in most cases in DEA K, which is the number of DMUs, is much larger than $m+n$, which is the number of inputs and outputs we use to determine their efficiency. Consequently, solving the dual problem is preferable in DEA. So the model can be solved in the following way.

First we present the dual problem. We use a real variable θ and a nonnegative vector variable $\underline{\lambda} = (\lambda_1, \ldots, \lambda_K)^T$, as variables. It takes the following form

$$
\begin{aligned}
(DLP_o) \quad & \min \theta \\
& s.t. \quad \sum_{k=1}^{K} \lambda_k y_{jk} \geq y_{jo} \;\; j = 1, \ldots, m \\
& \qquad\;\; \sum_{k=1}^{K} \lambda_k x_{ik} \leq \theta x_{io} \;\; i = 1, \ldots, n \\
& \qquad \theta \in \mathbb{R}, \;\; \lambda_1, \ldots, \lambda_K \;\geq 0\,,
\end{aligned}
$$

or equivalently, in matrix form,

$$
\begin{aligned}
(DLP_o) \quad & \min \theta \\
& s.t. \qquad Y\underline{\lambda} \geq \underline{y_o} \\
& \qquad\qquad X\underline{\lambda} \leq \theta \underline{x_o} \\
& \qquad \theta \in \mathbb{R}, \;\; \underline{\lambda} \;\geq \underline{0}\,.
\end{aligned}
$$

It can be easily observed that the solution $\theta = 1$, $\lambda_o = 1$, $\lambda_k = 0$, $k \neq o$ is feasible for any such problem. Thus, any optimal solution $\theta^\star$ is never greater than 1. The restriction for λ forces it to be non-zero, since we assumed the data, here $\underline{y_o}$, to be semi-positive. That forces $\theta^\star$ to be positive. So we have guarantied that $0 < \theta^\star \leq 1$.

The next step is to observe the relationship between the dual problem and the production possibility set P. We can see that the constraints in (DLP$_o$) require that the activity $(\theta \underline{x_o}, \underline{y_o})$ belongs to P, and the problem seeks to find the minimum θ that reduces (since $\theta \leq 1$) the input vector $\underline{x_o}$ proportionally, while remaining in the production possibility set. In other words, in (DLP$_o$) we are looking for an activity that maintains at least the level of every single output while reducing the input levels radially, as far as can be achieved, to $\theta \underline{x_o}$. Thus, when $\theta = 1$, the inputs cannot be feasibly decreased, so DMU$_o$ is considered (radially) efficient, while, when $\theta < 1$, the inputs can be radially decreased, so it is inefficient.

Under the constraints in the dual problem it can be easily inferred that the linear combination $(X\underline{\lambda}, Y\underline{\lambda})$ would be more efficient than $(\theta \underline{x_o}, \underline{y_o})$, since the first activity needs at most the same input and produces at least as much output. The input *excesses* $\underline{s}^- \in \mathbb{R}^n$ and output *shortfalls* $\underline{s}^+ \in \mathbb{R}^m$ that exist between them are identified as "slack" vectors, and defined as

$$
\underline{s}^- = \theta \underline{x_o} - X\underline{\lambda}, \;\; \underline{s}^+ = Y\underline{\lambda} - \underline{y_o},
$$

where $\underline{s}^- \geq \underline{0}$, $\underline{s}^+ \geq \underline{0}$ for any feasible solution $(\theta, \underline{\lambda})$ of (DLP$_o$).

To discover and calculate the possible input excesses and output shortfalls we solve the following two-phase linear programming problem.

Phase 1

We solve (DLP_o). Let $\theta^\star$ be the optimal solution. By the duality theorem of linear programming, that is also the (LP_o)'s value, and is therefore the CCR-efficiency value, also called "Farrell Efficiency" (after M.J.Farrell, 1957). This value is then incorporated into Phase 2.

Phase 2

Using $\theta^\star$, we solve the following LP

$$
\begin{aligned}
(LP_1) \quad \max w = \ & \underline{e}'\underline{s}^- + \underline{e}'\underline{s}^+ \\
s.t. \qquad & \underline{s}^- = \theta^\star \underline{x_o} - X\underline{\lambda} \\
& \underline{s}^+ = Y\underline{\lambda} - \underline{y_o} \\
& \underline{s}^- \geq \underline{0},\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0}\,,
\end{aligned}
$$

with variables $\underline{s}^-, \underline{s}^+$ and λ, and where $\underline{e} = (1, \ldots, 1)$, so that $\underline{e}'\underline{s}^- = \sum_{i=1}^n s_i^-$ and $\underline{e}'\underline{s}^+ = \sum_{j=1}^m s_i^+$. Equivalently,

$$
\begin{aligned}
(LP_1) \quad \max w = \ & \sum_{i=1}^{n} s_i^- + \sum_{j=1}^{m} s_j^+ \\
s.t. \qquad & s_i^- = \theta^\star x_{io} - \sum_{k=1}^{K} x_{ik}\lambda_k,\ i = 1, \ldots, n \\
& s_j^+ = \quad y_{jo} - \sum_{k=1}^{K} y_{jk}\lambda_k,\ j = 1, \ldots, m \\
& s_i^- \geq 0,\ s_j^+ \geq 0,\ \lambda_k \geq 0\,,
\end{aligned}
$$

So the objective of Phase 2 is to maximize the sum of excesses and shortfalls while keeping the CCR-efficiency $\theta^\star$ fixed and identify any possible slacks.

The objective function can be weighted instead, and still identify the existence of possible slacks, though they may have different value. For example the objective could be

$$
w = \underline{w_x}'\underline{s}^- + \underline{w_y}'\underline{s}^+,
$$

where $\underline{w_x} \in \mathbb{R}^n$ and $\underline{w_y} \in \mathbb{R}^m$ are row vectors.

Definition 3.1 Max-Slack Solution, Zero-Slack Activity An optimal solution $(\underline{\lambda}^\star, \underline{s}^{-\star}, \underline{s}^{+\star})$ of Phase 2 is called *max-slack solution*. If the max-slack solution satisfies $\underline{s}^{-\star} = 0$, and $\underline{s}^{+\star} = 0$, then it is called *zero-slack*.

Definition 3.2 CCR-Efficiency

If an optimal solution $(\theta^\star, \underline{\lambda}^\star, \underline{s}^{-\star}, \underline{s}^{+\star})$ of the two LPs above satisfies $\theta^\star = 1$ and is zero-slack, then the DMU_o is called *CCR-efficient*. Otherwise, it is called *CCR-inefficient*.

Note that

(i) $\theta^\star = 1$.

(ii) All slacks are zero.

must both be satisfied if full efficiency is to be attained.

The first of the two conditions is referred to as *ratio efficiency*. It is also referred to as *technical* or *weak efficiency*, since a value $\theta^\star < 1$ means that all inputs can be simultaneously reduced without altering the proportions (mix) in which they are utilized by DMU_o by the original inputs multiplied by $(1 - \theta^\star)$, to reach higher efficiency.

Any further reduction associated with non-zero slacks will change the input proportions. Hence the inefficiencies caused by non-zero slacks are referred to as *mix inefficiencies*.

Definition 3.3 Ratio and Mix Inefficiency

If a DMU has $\theta^\star < 1$, then it is *ratio inefficient*. If a DMU has non-zero slacks, then it is *mix inefficient*.

The conditions (i) and (ii) together describe the so-called "Pareto-Koopmans Efficiency".

Definition 3.4 Pareto-Koopmans Efficiency

A DMU is *fully efficient* if and only if it is not possible to improve any input or output without worsening some other input or output.

We will prove that the CCR-efficiency as defined above is the same as it was defined in Chapter 2 in Definition 2.1.

Theorem 3.1

The CCR-efficiency given in Definition 3.1 is equivalent to that given in Definition 2.1.

Proof Firstly, the vectors $\underline{u}$ and $\underline{v}$ are dual multipliers corresponding to the constraints (3.2) and (3.3) of (DLP$_o$), respectively. That means that $\underline{u}^\star\, \underline{s}^{-\star} = 0$ and $\underline{v}^\star\, \underline{s}^{+\star} = 0$.

Let DMU$_o$ be CCR-efficient by the Definition given in Chapter 2. Then $\theta^\star = 1$, so the first condition of Definition 3.2 is satisfied, and $\underline{u}^\star > 0$, $\underline{v}^\star > 0$, so by the complementary slackness conditions mentioned above, and since $\underline{s}^- \geq 0$ and $\underline{s}^+ \geq 0$, it can be inferred that $\underline{s}^{-\star} = 0$ and $\underline{s}^{+\star} = 0$. So all slacks are zero, and DMU$_o$ is also CCR-efficient by the second Definition.

To prove the other side of this equivalence, we have the first condition ready by definition, and need only prove the second. Using the Strong Theorem of Complementarity, since the slacks are all zero, we can infer that the corresponding variables are strictly positive. That means that, for DMU$_o$, $\theta^\star = 1$ and $\underline{u}^\star$, $\underline{v}^\star > 0$. So it is CCR-efficient by the first definition of CCR-efficiency, too.

So the CCR-efficiency as defined in Definition 2.1 and Definition 3.2 are equivalent. Solving the dual problem is preferable to the original for a few reasons. Some of them are presented below.

1. For computational purposes. In a linear programming problem as the number of constraints grow, so does, mostly, the computational complexity. Usually in DEA, the number of DMUs (K), which is the number of constraints, is considerably larger than the number of inputs and outputs $(n+m)$, so it is much less complex to solve the dual problem, which has $n+m$ constraints.

2. In the multiplier model, in the case of $\theta^\star = 1$, with either $\underline{u}_j^\star = 0$ or $\underline{v}_i^\star = 0$, in the definition for CCR-efficiency you cannot judge its efficiency using this solution, since you don't know if another solution exists with positive multipliers. In the case of the dual model, on the other hand, CCR-efficiency is defined for each different solution.

3. Using the simple linear problem, we can't easily compute the mix inefficiencies of a DMU, which consist of the input excess and output shortfalls. To compute them we create the second phase, which also results in the proper $\underline{\lambda}^\star$, so as to ascertain the DMU's peer group.

4. The interpretations of the dual problem are more straightforward since they are characterised as inputs and outputs while in the original problem the multipliers represent evaluations of the observed values. Those are also important in the evaluation but only as a secondary analysis of the results of the evaluation.

3.4 The Reference Set and Improvement in Efficiency

Definition 3.4 Reference Set

For an inefficient DMU$_o$, we define its *reference set* E_o, based on the max-slack solution as obtained in phases one and two, by

$$E_o = \{k \in \{1, 2, \ldots, K\} \mid \lambda_k > 0\}$$

.

An optimal solution can be expressed by

$$\theta^\star \underline{x_o} = \sum_{k=1}^{K} \lambda_k^\star \underline{x_k} + \underline{s}^{-\star}, \tag{3.1}$$

$$\underline{y_o} = \sum_{k=1}^{K} \lambda_k^\star \underline{y_k} - \underline{s}^{+\star}. \tag{3.2}$$

This can lead to the following formulae

$$\underline{x_o} \geq \theta^\star \underline{x_o} - \underline{s}^{-\star} = \sum_{k \in E_o} \lambda_k^\star \underline{x_k}, \tag{3.3}$$

$$\underline{x_o} \geq \text{technical} - \text{mix efficiency} \tag{3.4}$$

$$= \text{a positive combination of observed input values,} \tag{3.5}$$

and

$$\underline{y_o} \geq \underline{y_o} + \underline{s}^{+\star} = \sum_{k \in E_o} \lambda_k^\star \underline{y_k}, \tag{3.6}$$

$$\underline{y_o} \geq \text{observed outputs} + \text{shortfalls} \tag{3.7}$$

$$= \text{a positive combination of observed output values.} \tag{3.8}$$

The relations above can lead to the conclusion that the DMU's efficiency can be improved to the level of its peer group by reducing inputs and increasing outputs. Specifically, we should radially reduce the input values by $\theta^\star$ and eliminate the excess input $\underline{s}^{-\star}$. In much the same vein, it can be improved through an augmenting of the outputs, eliminating the shortfalls in $\underline{s}^{+\star}$. The total improvement in inputs, $\Delta \underline{x_o}$, and outputs, $\Delta \underline{y_o}$, is therefore

$$\Delta \underline{x_o} = \underline{x_o} - \left(\theta^\star \underline{x_o} - \underline{s}^{-\star}\right) = (1-\theta^\star)\,\underline{x_o} + \underline{s}^{-\star}, \tag{3.9}$$

$$\Delta \underline{y_o} = \underline{s}^{+\star}. \tag{3.10}$$

Using this formula to improve a DMU's efficiency we get the new inputs and outputs

$$\underline{\hat{x}_o} = \underline{x_o} - \Delta \underline{x_o} = \theta^\star \underline{x_o} - \underline{s}^{-\star} \leq \underline{x_o}, \tag{3.11}$$

$$\underline{\hat{y}_o} = \underline{y_o} + \Delta \underline{y_o} = \underline{y_o} + \underline{s}^{+\star} \geq \underline{y_o}. \tag{3.12}$$

Formulas (3.11) and (3.12) are called *CCR projection*. It is not the only formula for improvement. We will see in the next section that the improved activity $(\underline{\hat{x}_o}, \underline{\hat{y}_o})$ is a projection of DMU$_o$ into its reference set E_o.

Theorem 3.2

The improved activity $(\underline{\hat{x}_o}, \underline{\hat{y}_o})$ defined by (3.19) and (3.20) is CCR-efficient.

Proof We will evaluate the efficiency of $(\underline{\hat{x}_o}, \underline{\hat{y}_o})$.

$$(DLP_o) \quad \max\ \theta \tag{3.13}$$

$$s.t.\ \ \theta \underline{\hat{x}_o} - X\underline{\lambda} - \underline{s}^- \qquad = \underline{0} \tag{3.14}$$

$$Y\underline{\lambda} \qquad - \underline{s}^+ = \underline{\hat{y}_o} \tag{3.15}$$

$$\theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},\ \underline{s}^- \geq \underline{0},\ \underline{s}^+ \geq \underline{0}. \tag{3.16}$$

Let $(\hat{\theta}, \underline{\hat{\lambda}}, \underline{\hat{s}}^-, \underline{\hat{s}}^+)$ be an optimal solution of (DLP_o). Then, inserting the formulas (3.11) and (3.12) into (3.14) and (3.15), we get the following

$$\begin{aligned} \hat{\theta}\theta^\star \underline{x_o} &= X\underline{\hat{\lambda}} + \underline{\hat{s}}^- + \hat{\theta}\underline{s}^{-\star} \\ \underline{y_o} &= Y\underline{\hat{\lambda}} - \underline{\hat{s}}^+ - \underline{s}^{+\star}. \end{aligned}$$

The solution can also be written in the following way

$$\begin{aligned} \tilde{\theta}\underline{x_o} &= X\underline{\hat{\lambda}} + \underline{\tilde{s}}^- \\ \underline{y_o} &= Y\underline{\hat{\lambda}} - \underline{\tilde{s}}^+, \end{aligned}$$

where $\tilde{\theta} = \hat{\theta}\theta^\star$, $\underline{\tilde{s}}^- = \underline{\hat{s}}^- + \hat{\theta}\underline{s}^{-\star}$ and $\underline{\tilde{s}}^+ = \underline{\hat{s}}^+ + \underline{s}^{+\star}$. Since $\theta^\star$ is an optimal solution, it follows that $\tilde{\theta} = \hat{\theta}\theta^\star = \theta^\star$, so $\hat{\theta} = 1$. Also, since $\underline{es}^{-\star} + \underline{es}^{+\star}$ is maximal,

$$\underline{e}'\underline{\tilde{s}}^- + \underline{e}'\underline{\tilde{s}}^+ = (\underline{e}'\underline{\hat{s}}^- + \underline{e}'\underline{s}^{-\star}) + (\underline{e}'\underline{\hat{s}}^+ + \underline{e}'\underline{s}^{+\star}) \leq \underline{e}'\underline{s}^{-\star} + \underline{e}'\underline{s}^{+\star}$$

Consequently, $\underline{e}'\underline{\hat{s}}^- + \underline{e}'\underline{\hat{s}}^+ = 0$ and so $\underline{\hat{s}}^- = \underline{\hat{s}}^+ = \underline{0}$. Hence, all the conditions of Definition 3.2 are satisfied, and the improved activity is CCR-efficient.

Corollary 3.1

The point $(\underline{\hat{x}_o}, \underline{\hat{y}_o})$ defined by (3.19) and (3.20) is the point on the efficient frontier used to evaluate the performance of DMU_o.

The above Corollary shows that DMU_o's input excesses are equal to $\underline{x_o} - \underline{\hat{x}_o}$ and output shortfalls as $\underline{\hat{y}_o} - \underline{y_o}$.

Theorem 3.3

The DMUs in the reference set, E_o, are CCR-efficient, and so is any semi-positive combination.

3.5 Example: CCR, input-oriented

Example 3.1

Say, for example, that you have 10 different locations of a chain of coffeehouses. In each, the store has complete managerial freedom and are evaluated at the end of the year using the DEA approach on their sales and on customer satisfaction (on a scale of 1 to 10), when compared to the money they were provided with at the start of the year. In the following table (3.1) we present their results from last year.

Locations	A	B	C	D	E	F	G	H	I	J
Money provided	10	3	5	9	7	6	8	2	4	8
Sales	17	9	29	12	17	29	30	8	10	25
Customer satisfaction	10	15	10	10	1	6	4	10	9	9

Table 3.1: Single Input and Two Outputs Case

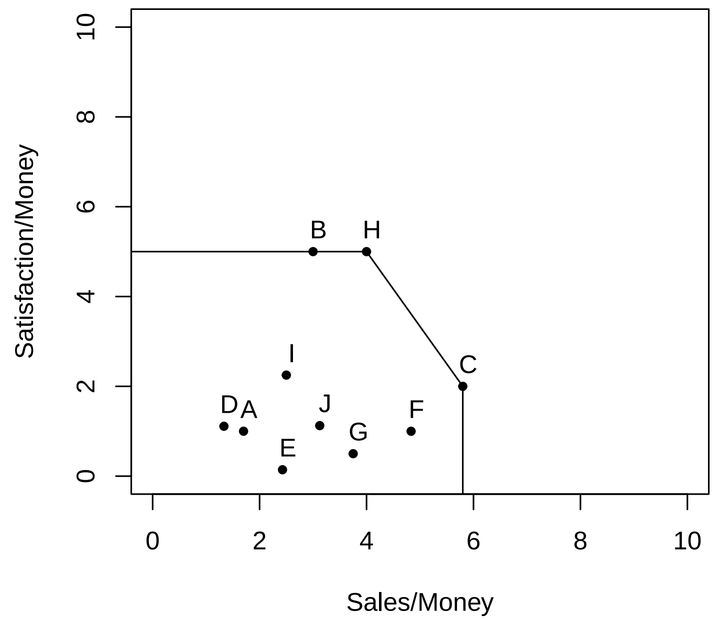


Figure 3.2: Sales and Satisfaction per Money permitted

Here, the first Phase of 2-Phased dual problem for C takes the form

$$\begin{aligned} < C > \quad & \min\ \theta \\ s.t.\ & 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \geq 29 \\ & 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 10 \\ & 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 5\theta \\ & \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0}, \end{aligned}$$

where $\theta^\star = 1$, and the second phase becomes

$$\begin{aligned} < C > \quad & \max\ w = s^- + s_1^+ + s_2^+ \\ & s.t.\ s_1^+ = -29 + 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \\ & \quad\ s_2^+ = -10 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\ & \quad\ s^- = 5 - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\ & \quad\ s^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0}, \end{aligned}$$

where $s^{-\star} = 0$, $s_1^{+\star} = 0$ and $s_2^{+\star} = 0$, $\lambda_3^\star = 1$ and $\lambda_k^\star = 0$ for every $k = 1, 2, 4, \ldots, 10$ and therefore the DMU C is CCR-efficient with the second Definition, as can be seen in Figure (3.2).

Doing the same for another DMU, say B, we get

$$\begin{aligned}
< B > \quad & \min \theta \\
s.t.\ & 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \geq 9 \\
& 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 15 \\
& 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 3\theta \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},
\end{aligned}$$

where $\theta^\star = 1$, $\lambda_2^\star = 1$ and $\lambda_k^\star = 0$ for every $k = 1, 3, \ldots, 10$, and the second phase becomes

$$\begin{aligned}
< B > \quad & \max w = s^- + s_1^+ + s_2^+ \\
& s.t.\ s_1^+ = -9 + 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \\
& s_2^+ = -15 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\
& s^- = 3 - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\
& s^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0},
\end{aligned}$$

where, as expected, we get positive slacks $s^{-\star} = 0$, $s_1^{+\star} = 3$ and $s_2^{+\star} = 0$, $\lambda_8^\star = 1.5$ and $\lambda_k^\star = 0$ for every $k = 1, 2, 4, \ldots, 10$ so while B is radially efficient, it is not fully efficient, and his peer group consists of H and not B. The improved activity would be

$$\begin{aligned}
x_B = \theta^\star x_2 - s^{-\star} = 3 - 0 = 3 = 1.5 \times 2 &= \sum_{k=1}^{K} \lambda_k^\star x_k \\
y_{B1} = y_{21} + s_1^{+\star} = 9 + 3 = 12 = 1.5 \times 8 &= \sum_{k=1}^{K} \lambda_k^\star y_{1k} \\
y_{B2} = y_{22} + s_2^{+\star} = 15 + 0 = 15 = 1.5 \times 10 &= \sum_{k=1}^{K} \lambda_k^\star y_{2k}.
\end{aligned}$$

Lastly, if we try the same for A,

$$\begin{aligned}
< A > \quad & \min \theta \\
s.t.\ & 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \geq 17 \\
& 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 10 \\
& 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 10\theta \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},
\end{aligned}$$

where $\theta^\star = 0.329$, $\lambda_3^\star = 0.4$, $\lambda_8^\star = 0.6$ and $\lambda_k^\star = 0$ for every $k = 1, 2, 4, \ldots, 7, 9, 10$, and the second phase becomes

$$
\begin{aligned}
< A > \quad & \max\ w = s^- + s_1^+ + s_2^+ \\
& s.t.\ s_1^+ = -17 + 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \\
& \quad\ s_2^+ = -10 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\
& \quad\ s^- = 32.9 - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\
& \quad\ s^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0},
\end{aligned}
$$

and the slacks are $s^{-\star} = 0$, $s_1^{+\star} = 0$ and $s_2^{+\star} = 0$, with $\lambda_3^\star = 0.4$, $\lambda_8^\star = 0.6$, approximately, and $\lambda_k^\star = 0$, otherwise. So this DMU exhibits radial inefficiencies. The improved activity would therefore be

$$
\begin{aligned}
\hat{x}_A &= \theta^\star x_1 - s^{-\star} = 3.29 - 0 = 3.29 \approx 0.4 \times 5 + 0.6 \times 2 = \sum_{k=1}^{K} \lambda_k^\star x_k \\
\hat{y}_{A1} &= y_{11} + s_1^{+\star} = 17 + 0 = 17 \approx 0.4 \times 29 + 0.6 \times 8 = \sum_{k=1}^{K} \lambda_k^\star y_{1k} \\
\hat{y}_{A2} &= y_{12} + s_2^{+\star} = 10 + 0 = 10 \approx 0.4 \times 10 + 0.6 \times 10 = \sum_{k=1}^{K} \lambda_k^\star y_{2k}.
\end{aligned}
$$

Doing so for every DMU gives us the following CCR-efficiencies (Table 3.2).

DMU	A	B	C	D	E	F	G	H	I	J
θ	0.33	1	1	0.29	0.42	0.83	0.65	1	0.55	0.54
s^-	0	0	0	0	0	0	0	0	0	0
s_1^+	0	3	0	0	0	0	0	0	0	0
s_2^+	0	0	0	0	4.9	4	6.3	0	0	0
λ_1	0	0	0	0	0	0	0	0	0	0
λ_2	0	0	0	0	0	0	0	0	0	0
λ_3	0.43	0	1	0.19	0.59	1	1.03	0	0.13	0.85
λ_4	0	0	0	0	0	0	0	0	0	0
λ_5	0	0	0	0	0	0	0	0	0	0
λ_6	0	0	0	0	0	0	0	0	0	0
λ_7	0	0	0	0	0	0	0	0	0	0
λ_8	0.57	1.5	0	0.81	0	0	0	1	0.77	0.05
λ_9	0	0	0	0	0	0	0	0	0	0
λ_{10}	0	0	0	0	0	0	0	0	0	0

Table 3.2: The CCR-efficiency

It is evident in this Table that the only DMUs that are included in the peer groups are C and H, which seeing the figure 3.2 would make sense. And the improved activities when compared to the original are presented in he following Table 3.3.

DMU	A	B	C	D	E	F	G	H	I	J
x	10	3	5	9	7	6	8	2	4	8
y_1	17	9	29	12	17	29	30	8	10	25
y_2	10	15	10	10	1	6	4	10	9	9
$\hat{x}$	3.3	3	5	2.6	2.9	5	5.2	2	2.2	4.3
$\hat{y}_1$	17	12	29	12	17	29	30	8	10	25
$\hat{y}_2$	10	15	10	10	5.9	10	10.3	10	9	9

Table 3.3: Original and Improved Activities

So, DMUs like B that have a CCR-efficiency of 1, can still be improved, in this case by increasing y_1 by 3. That is to say, it has only mix technical inefficiencies that in the original LP model, or only the first phase, would not be easily evident.

3.6 The Output-oriented CCR Model

So far, we have seen the input-oriented model, where

- in the LP, we aim to maximize weighted output while keeping the weighted input fixed,
- when there is ratio inefficiency, $\theta^\star < 1$, we fix it by multiplying inputs by $\theta^\star$, i.e. by reducing the inputs.

Thus, in input-oriented models we "control" inputs. However, in many cases, inputs are not easy to control or maybe the outputs are simply easier to control. In these cases, we use output-oriented models where

- in the LP, we aim to minimize weighted input while keeping the weighted output fixed,
- when there is ratio inefficiency, we fix it by increasing outputs.

First, the output-oriented problem, (LPO_o), can be expressed using the vectors $\underline{p} \in \mathbb{R}^n$ and $\underline{q} \in \mathbb{R}^m$ as follows.

$$\begin{aligned} (LPO_o) \quad & \max \underline{p}'\underline{x_o} \\ s.t. \quad & \underline{q}'\underline{y_o} = 1 \\ & -\underline{p}X + \underline{q}Y \le \underline{0} \\ & \underline{p} \ge \underline{0},\ \underline{q} \ge \underline{0}. \end{aligned}$$

The first and second phases of the corresponding dual LP can be written as follows.

Phase I

$$\begin{aligned} (DLPO_{o1}) \quad & \max \eta \\ s.t.\ & X\underline{\mu} \le \underline{x_o} \\ & Y\underline{\mu} \ge \eta\underline{y_o} \\ & \eta \in \mathbb{R},\ \underline{\mu} \ge \underline{0}\,. \end{aligned}$$

Phase II

$$\begin{aligned} (DLPO_{o2}) \quad & \max \underline{e}'\underline{t}^+ + \underline{e}'\underline{t}^- \\ s.t.\ & \underline{t}^- = \underline{x_o} - X\underline{\mu} \\ & \underline{t}^+ = -\eta\underline{y_o} + Y\underline{\mu}, \underline{\mu}, \underline{t}^+, \underline{t}^- \ge \underline{0}\,. \end{aligned}$$

An optimal solution of the output-oriented model can be derived from the input-oriented model's solution through the following.

$$\underline{\lambda} = \frac{\underline{\mu}}{\eta}, \quad \theta = \frac{1}{\eta}.$$

By replacing the variables in $(DLPO_o)$ as shown above, we get exactly (DLP_o), the input-oriented model. That means that, since $\theta^\star \le 1$, $\eta^\star \ge 1$. It stands to reason, then, to conclude that the higher the value of $\eta^\star$ for a DMU, the *less* efficient it is. This happens because, while $\theta^\star$ expresses the input reduction rate, $\eta^\star$ expresses the output enlargement rate.

In the output-oriented model, the slack, $(\underline{t}^-, \underline{t}^+)$, is defined by

$$\begin{aligned} X\underline{\mu} + \underline{t}^- &= \underline{x_o} \\ Y\underline{\mu} - \underline{t}^+ &= \eta\underline{y_o}, \end{aligned}$$

and is also related to the input-oriented model, via

$$\underline{t}^{-\star} = \frac{\underline{s}^{-\star}}{\theta^\star}, \quad \underline{t}^{+\star} = \frac{\underline{s}^{+\star}}{\theta^\star}. \tag{3.17}$$

From the duality of the two optimal solutions, we can conclude that a DMU will be CCR-efficient using the input-oriented model if and only if it is found to be CCR-efficient using the output-oriented model.

Therefore, the multipliers in the output-oriented model can be obtained from

$$\underline{p}^\star = \frac{\underline{v}^\star}{\theta^\star} \; \underline{q}^\star = \frac{\underline{u}^\star}{\theta^\star},$$

since $\underline{p}^\star \underline{x}_o = \frac{\underline{v}^\star \underline{x}_o}{\theta^\star} = \eta^\star$.

Following this logic, the improvement that can be obtained using this model is

$$\begin{aligned} \underline{\hat{x}_o} &= \underline{x}_o - \underline{t}^{-\star}, \\ \underline{\hat{y}_o} &= \eta^\star \underline{y}_o + \underline{t}^{+\star}. \end{aligned}$$

3.7 Example: CCR, output-oriented

Example 3.2 Using the data and premise of Example 3.1 we will present and solve the corresponding output-oriented model.

$$\begin{aligned} < B > \quad & \max \eta \\ s.t. \; & 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \geq 9\eta \\ & 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \geq 15\eta \\ & 10\mu_1 + 3\mu_2 + 5\mu_3 + 9\mu_4 + 7\mu_5 + 6\mu_6 + 8\mu_7 + 2\mu_8 + 4\mu_9 + 8\mu_{10} \leq 3 \\ & \eta \in \mathbb{R}, \; \underline{\mu} \geq \underline{0}, \end{aligned}$$

we get $\eta^\star = 1 = \frac{1}{\theta^\star}$. The second phase would then be

$$\begin{aligned} < B > \quad & \max w = t^- + t_1^+ + t_2^+ \\ s.t. \; & t_1^+ = -9 + 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \\ & t_2^+ = -15 + 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \\ & t^- = 3 - 10\mu_1 - 3\mu_2 - 5\mu_3 - 9\mu_4 - 7\mu_5 - 6\mu_6 - 8\mu_7 - 2\mu_8 - 4\mu_9 - 8\mu_{10} \\ & t^- \geq 0, \; \underline{t}^+ \geq \underline{0}, \; \underline{\mu} \geq \underline{0}, \end{aligned}$$

where the slacks, as expected from (3.10), are all zero except for $t_1^{+\star} = 3$, also $\mu_8^\star = 1.5$, and $\mu_k^\star = 0$, for all other k, like in the input-oriented model, since $\theta^\star = 1$.

In the case of A,

$$\begin{aligned} < A > \quad & \max\ \eta \\ s.t.\ & 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \geq 17\eta \\ & 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \geq 10\eta \\ & 10\mu_1 + 3\mu_2 + 5\mu_3 + 9\mu_4 + 7\mu_5 + 6\mu_6 + 8\mu_7 + 2\mu_8 + 4\mu_9 + 8\mu_{10} \leq 10 \\ & \eta \in \mathbb{R},\ \underline{\mu} \geq \underline{0}, \end{aligned}$$

the solution has $\eta^\star = 3.04 = \frac{1}{\theta^\star}$. As for the second phase,

$$\begin{aligned} < A > \quad & \max\ w = t^- + t_1^+ + t_2^+ \\ s.t.\ & t_1^+ = -51.68 + 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \\ & t_2^+ = -30.4 + 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \\ & t^- = 10 - 10\mu_1 - 3\mu_2 - 5\mu_3 - 9\mu_4 - 7\mu_5 - 6\mu_6 - 8\mu_7 - 2\mu_8 - 4\mu_9 - 8\mu_{10} \\ & t^- \geq 0,\ \underline{t}^+ \geq \underline{0},\ \underline{\mu} \geq \underline{0}, \end{aligned}$$

we get, like in the input-oriented model, zero slacks, while $\mu_3^\star = 1.3$, $\mu_8^\star = 1.7$, and $\mu_k^\star = 0$, for all other k.

In all, the results were as expected from the transformations of the solutions found on the input-oriented model in Example 3.1.

3.8 Conclusion

In this chapter we have relaxed the positive data assumption to a semi-positive one. Doing so, we have further analysed the input-oriented model by presenting the dual problem and further enriching our results by the use of a second phase that can give us any possible input excesses or output shortfalls. Succeeding that, we redefined the terms CCR-efficiency, Pareto-Koopmans efficiency and reference set, using the results of that two-phased model, and showed that they are, in fact, equivalent to the previously defined ones.

Finally, we presented the output-oriented model, which maximizes the DMU's virtual output while keeping its virtual input fixed. It can be easily shown that the value and the solutions of the output-oriented model can be inferred by the ones of the input-oriented model, and vice-versa.

Chapter 4

An Alternative DEA Model

4.1 Introduction

So far we have analysed the CCR model, which assumes that for any activity in the possibility set, any activity that has the same proportion of inputs and outputs also belongs to the production possibility set. That is to say, it has a constant returns-to-scale assumption. By expanding that assumption to increasing and decreasing returns-to-scale other models can be introduced, chief of which is the BCC (Banker-Charnes-Cooper) model, which assumes a production possibility set spanned by convex hull of the existing DMUs.

4.2 The BCC Model

Banker, Charnes and Cooper (1984) [1] defined the production possibility set as

$$P_B = \{(\underline{x}, \underline{y}) |\, \underline{x} \geq X\underline{\lambda},\, \underline{y} \leq Y\underline{\lambda},\, \underline{e}\underline{\lambda} = 1,\, \underline{\lambda} \geq \underline{0}\},$$

where $X = (x_i)$ and $Y = (y_j)$ are the data set, and $\underline{e}$ is a vector of n ones.

The constraint $\underline{e}\underline{\lambda} = 1$, which imposes the convexity of the production possibility set, is where the BCC model differs to the CCR one. Since it is the only difference between them, the feasible region of the BCC model, P_B, is a subset of the CCR models feasible region, so the BCC's efficiency score can be no less than the CCR's one.

To further illustrate what this constraint means for the efficiency evaluation, we will present an example of the simple single input - single output case.

Example 4.1

We have 8 DMUs that have the inputs and outputs shown in the Figure 4.1, where the CCR efficient frontier would be spanned by the line connecting B with the origin, and the BCC one can be seen in the figure. It is the line connecting F, B and A and can be built as the boundary of the production possibility set, as defined above.

In Figure 4.2 we can see the efficient frontier of the CCR model, in a dotted line. In the input-oriented model, by keeping the output fixed and projecting G to the efficient frontier, we can calculate the CCR-efficiency and the BCC-efficiency of DMU G, by the ratios

$$\theta_{CCR} = \frac{\frac{\text{output}_\text{G}}{\text{input}_\text{G}}}{\frac{\text{output}_\text{R}}{\text{input}_\text{R}}} = \frac{\text{input}_\text{R}}{\text{input}_\text{G}} = \frac{PR}{PG} = 0.267,$$

$$\theta_{BCC} = \frac{\frac{\text{output}_\text{G}}{\text{input}_\text{G}}}{\frac{\text{output}_\text{Q}}{\text{input}_\text{Q}}} = \frac{\text{input}_\text{Q}}{\text{input}_\text{G}} = \frac{PQ}{PG} = 0.8.$$

In general, CCR-efficiency will never be more than BCC-efficiency, cause the efficient frontier is closer to all the units in the BCC model.

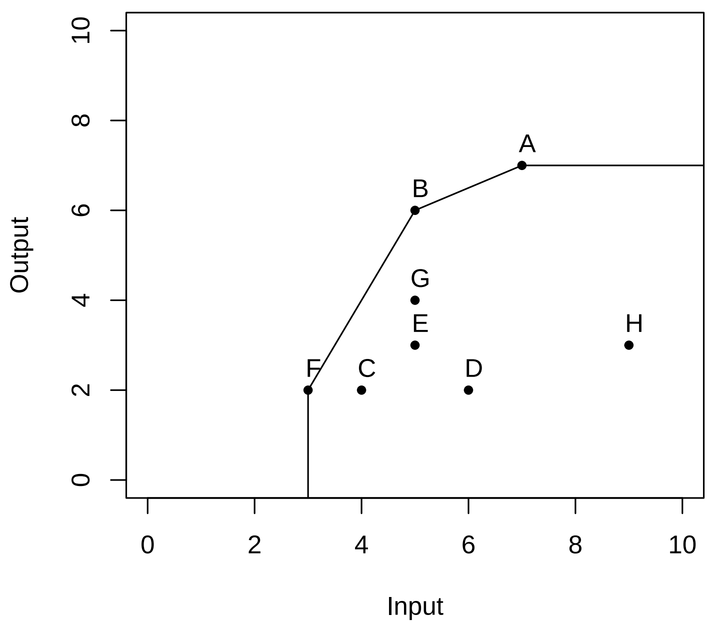


Figure 4.1: Plot of Output to Input

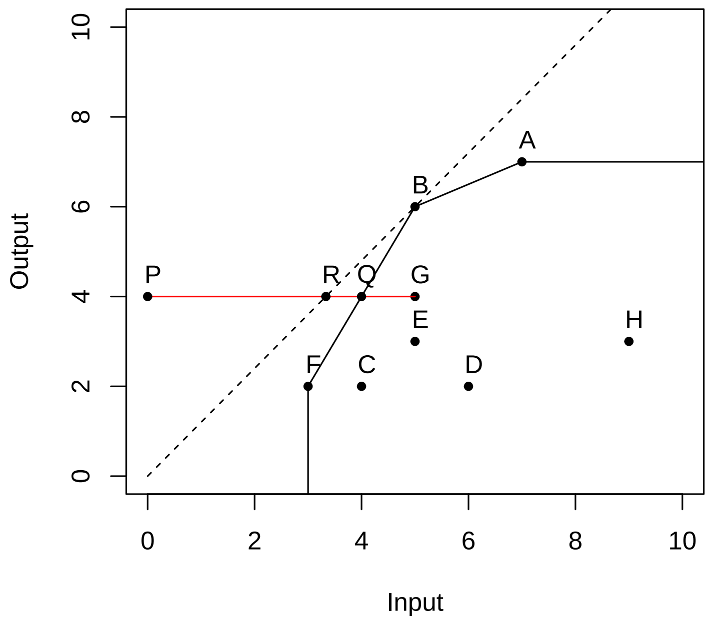


Figure 4.2: CCR and BCC efficiency of G (input-oriented)

Using the output-oriented BCC model in the same example the Figure 4.2 is inverted and the efficiency will be evaluated as follows. (4.3)

$$\theta_{CCR} = \frac{\frac{\text{output}_\text{G}}{\text{input}_\text{G}}}{\frac{\text{output}_\text{B}}{\text{input}_\text{B}}} = \frac{\text{output}_\text{G}}{\text{output}_\text{B}} = \frac{TG}{TB} = 0.67.$$

$$\theta_{BCC} = \frac{\frac{\text{output}_\text{G}}{\text{input}_\text{G}}}{\frac{\text{output}_\text{B}}{\text{input}_\text{B}}} = \frac{\text{output}_\text{G}}{\text{output}_\text{B}} = \frac{TG}{TB} = 0.67.$$

In general, the evaluation of a DMU's efficiency will always be better using the BCC model, as can be seen above.

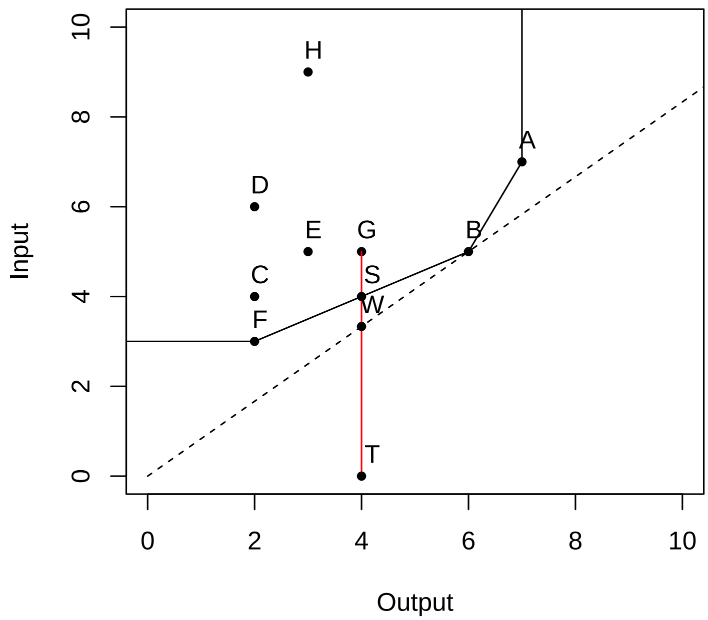


Figure 4.3: CCR and BCC efficiency of G (output-oriented)

The input-oriented model in its general form is the following.

$$
\begin{aligned}
(BCC_o)\ \min \theta_B & \\
s.t. \quad & Y\underline{\lambda} \geq \underline{y_o} \\
& X\underline{\lambda} \leq \theta_B \underline{x_o} \\
& \underline{e}'\underline{\lambda} = 1 \\
\theta_B \in \mathbb{R}, \quad & \underline{\lambda} \geq \underline{0}\,,
\end{aligned}
$$

where θ_B is a scalar. Following that we implement the second phase, which is simply a copy of the second phase in the CCR dual model, along with the added constraint of $\underline{e}'\underline{\lambda} = 1$, to get the slacks $\underline{s}^{-\star}$ and $\underline{s}^{+\star}$.

The dual modifier form of this is the following.

$$
\begin{aligned}
(BCC_1)\ \max z = \quad & \underline{u}'\underline{y_o} - u_o \\
s.t. \quad \underline{v}'\underline{x_o} \quad & = 1 \\
-\underline{v}X + \underline{u}Y - u_o\underline{e} & \leq \underline{0} \\
\underline{v} \geq \underline{0},\ \underline{u} \geq \underline{0},\ u_o & \in \mathbb{R},
\end{aligned}
$$

where $\underline{u}$ and $\underline{v}$ are vectors and z and u_o are scalars (u_o can be negative or zero).

To better understand the similarities and differences between the BCC and CCR models we will also present the corresponding fractional problem.

$$
\begin{aligned}
(BCC_2)\max z &= \frac{\underline{u}'\underline{y_o} - u_o}{\underline{v}'\underline{x_o}} \\
s.t. \quad & \frac{\underline{u}'\underline{y_k} - u_o}{\underline{v}'\underline{x_k}} \leq 1, \;\; k = 1, \dots, K \\
& \underline{v} \geq \underline{0},\; \underline{u} \geq \underline{0},\; u_o \in \mathbb{R}.
\end{aligned}
$$

The difference between the two fractional problems is solely the variable u_o which is the variable corresponding to the convexity constraint and allows for the use of negative outputs by way of adding or subtracting a variable to the outputs (different for each DMU$_o$).

Definition 4.1 BCC-Efficiency

If an optimal solution $(\theta^\star, \underline{\lambda}^\star, \underline{s}^{-\star}, \underline{s}^{+\star})$ of the 2-phase problem above satisfies $\theta^\star = 1$ and has zero slacks, then the DMU$_o$ is called *BCC-efficient*. In any other case, it is called *BCC-inefficient*.

Definition 4.2 Reference Set

For any BCC-inefficient DMU, we define their *reference set*, based on the optimal solution $\underline{\lambda}^\star$, as follows

$$E_o = \{k \in 1, \dots, K \,|\, \lambda_k^\star > 0\}.$$

Following that, we can improve the BCC-inefficient DMUs via *BCC-projection* to

$$
\begin{aligned}
\underline{\hat{x}_o} &= \theta_B^\star \underline{x_o} - \underline{s}^{-\star} \\
\underline{\hat{y}_o} &= \underline{y_o} + \underline{s}^{+\star}
\end{aligned}
$$

The following theorems are corresponding to the CCR ones.

Theorem 4.1

The improved activity $(\underline{\hat{x}_o}, \underline{\hat{y}_o})$ is BCC-efficient.

Theorem 4.2

Every DMU in E_o, associated then with a $\lambda_k^\star > 0$, as defined in Definition 4.2, is BCC-efficient.

And lastly, the following theorem corresponds exclusively to the input-oriented BCC model, and not the CCR model.

Theorem 4.3

Any DMU that has a minimum input value or a maximum output value for any item, is BCC-efficient.

In the interest of comparing the BCC and the CCR models and demonstrating the fact that the BCC is a softer approach to the evaluation of a DMU's efficiency, we will use the same example (Example 3.1) used in the CCR model and compare the results.

Example 4.2

Using the premise of Example 3.1, we present again the data in Table 4.1.

Locations	A	B	C	D	E	F	G	H	I	J
Money provided	10	3	5	9	7	6	8	2	4	8
Sales	17	9	29	12	17	29	30	8	10	25
Customer satisfaction	10	15	10	10	1	6	4	10	9	9

Table 4.1: Single Input and Two Outputs Case

Then, the input-oriented BCC model for, say, C is as follows.

$$
\begin{aligned}
<C> \quad & \min \theta \\
s.t.\ & 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \geq 29 \\
& 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 10 \\
& 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 5\theta \\
& \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} = 1 \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},
\end{aligned}
$$

The $\theta^\star$ here is 1, so moving on to the second phase, we have

$$
\begin{aligned}
<C> \quad & \max w = s^- + s_1^+ + s_2^+ \\
s.t.\ & s_1^+ = -29 + 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \\
& s_2^+ = -10 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\
& s^- = 5 - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\
& 1 = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} \\
& s^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0},
\end{aligned}
$$

which also gets the same result of $s^{-\star} = 0$, $s_1^{+\star} = 0$ and $s_2^{+\star} = 0$. Doing the same for all DMUs we get the following BCC-efficiencies (Table 4.2), which when compared to the CCR-efficiencies may not be tremendously different, but are generally bigger, while having one extra efficient DMU (G).

Locations	A	B	C	D	E	F	G	H	I	J
BCC-efficiencies	0.33	1	1	0.29	0.47	0.83	1	1	0.57	0.55
CCR-efficiencies	0.33	1	1	0.29	0.42	0.83	0.65	1	0.55	0.54

Table 4.2: The BCC-efficiency

4.3 The Output-oriented BCC Model

As with CCR, there is also an output-oriented BCC model. It takes the following form

$$\begin{aligned}(BCCO_o)\max \eta_B& \\ s.t.\quad Y\underline{\mu} &\geq \eta_B \underline{y_o} \\ X\underline{\mu} &\leq \underline{x_o} \\ \underline{e}'\underline{\mu} &= 1 \\ \eta_B \in \mathbb{R},\quad \underline{\mu} &\geq \underline{0},\end{aligned}$$

and the second phase is as it is in CCR, with the convex constraint for μ added. An optimal solution of $\eta_B = 1$, with zero slacks indicates a BCC-efficient DMU, and any other solution indicates a BBC-inefficient one. Note that, while that is true, the BCC efficiency score of the output-oriented model is not necessarily the inverted input-oriented one, like in the CCR model.

The modifier form, its dual, can be expressed as:

$$\begin{aligned}(BCCO_1)\max z = \underline{v}'\underline{x_o} \qquad &- v_o \\ s.t.\qquad \underline{u}'\underline{y_o} \qquad &= 1 \\ \underline{v}X - \underline{u}Y - v_o\underline{e} &\geq \underline{0} \\ \underline{v} \geq \underline{0},\ \underline{u} \geq \underline{0},\ v_o &\in \mathbb{R}.\end{aligned}$$

An example of this output-oriented model, would be to take the example above (Example 4.2) and solve it as an output-oriented BCC model.

Example 4.3

In the spirit of observing the differences, we will once again judge the efficiency of the DMU C. The problem is as follows.

$$
\begin{aligned}
< C > \quad & \max \eta_B \\
s.t.\ \ & 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \geq 29\eta_B \\
& 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \geq 10\eta_B \\
& 10\mu_1 + 3\mu_2 + 5\mu_3 + 9\mu_4 + 7\mu_5 + 6\mu_6 + 8\mu_7 + 2\mu_8 + 4\mu_9 + 8\mu_{10} \leq 5 \\
& \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 + \mu_9 + \mu_{10} = 1 \\
& \eta_B \in \mathbb{R},\ \underline{\mu} \geq \underline{0},
\end{aligned}
$$

with $\eta_B^\star = 1$. The second phase will then take the form

$$
\begin{aligned}
< C > \quad & \max w = t^- + t_1^+ + t_2^+ \\
s.t.\ \ & t_1^+ = -29 + 17\mu_1 + 9\mu_2 + 29\mu_3 + 12\mu_4 + 17\mu_5 + 29\mu_6 + 30\mu_7 + 8\mu_8 + 10\mu_9 + 25\mu_{10} \\
& t_2^+ = -10 + 10\mu_1 + 15\mu_2 + 10\mu_3 + 10\mu_4 + \mu_5 + 6\mu_6 + 4\mu_7 + 10\mu_8 + 9\mu_9 + 9\mu_{10} \\
& t^- = 5 - 10\mu_1 - 3\mu_2 - 5\mu_3 - 9\mu_4 - 7\mu_5 - 6\mu_6 - 8\mu_7 - 2\mu_8 - 4\mu_9 - 8\mu_{10} \\
& 1 = \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 + \mu_9 + \mu_{10} \\
& t^- \geq 0,\ \underline{t}^+ \geq \underline{0},\ \underline{\mu} \geq \underline{0},
\end{aligned}
$$

where $t^{-\star} = 0$, $t_1^{+\star} = 0$ and $t_2^{+\star} = 0$, and where $\mu_3 = 1$ and $\mu_k = 0$ for all other k.

The results for the η for the CCR and the BCC output-oriented model are presented in Table 4.3.

Locations	A	B	C	D	E	F	G	H	I	J
BCC-efficiencies	1.21	1	1	1.33	1.75	1.01	1	1	1.5	1.13
CCR-efficiencies	3	1	1	3.45	2.38	1.2	1.54	1	1.82	1.85

Table 4.3: The BCC-efficiency

So we can easily observe that, while for the CCR output-oriented model, the efficiency scores are the input-oriented ones inverted, that is not the case for the BCC model.

Chapter 5

Alterations on the DEA Models

5.1 Introduction

In this chapter we will present models that are more specialised to each occasion. That mainly encompasses cases in which inputs are out of managerial control, and should be taken into account in a different way, and cases where categorical variables come into play.

5.2 Discretionary and Non-Discretionary Inputs

In the previous analysis of the DMUs' efficiency we assumed that all inputs and outputs can be affected by managerial decisions. In truth, that is not always the case. There are some variables that affect the DMU's efficiency but are not under the management's control. Examples of this are the weather patterns that affect airplane flights, or the initial capital of a business.

The ones that can be controlled by using different managerial methods are called *Discretionary* while the rest are called *Non-Discretionary*.

Even though they cannot be changed, it is important to take them into account when evaluating a DMU's efficiency, as they affect their efficiency. Of course, since they cannot be improved, they will not be taken into account when it comes to possible improvements a DMU can make.

Banker and Morey (1986) [2] referred to those variables as *exogenously fixed* and evaluated the efficiency of DMUs in presence of exogenously fixed inputs by the following problem.

$$
\begin{aligned}
\min \theta - \epsilon \left(\sum_{i \in D} s_i{}^- + \sum_{j=1}^{m} s_j{}^+ \right) \\
s.t.\ \theta x_{io} &= \sum_{k=1}^{K} x_{ik}\lambda_k + s_i{}^-, \;\; i \in D \\
x_{io} &= \sum_{k=1}^{K} x_{ik}\lambda_k + s_i{}^-, \;\; i \in ND \\
y_{jo} &= \sum_{k=1}^{K} y_{jk}\lambda_k - s_j{}^+, \;\; j = 1, \ldots, m \\
\theta \in \mathbb{R}, s_i{}^-, s_j{}^+, \lambda_k \geq 0.
\end{aligned}
$$

where D and ND refer to the sets of discretionary and non-discretionary inputs, respectively. We do not apply θ to the non-discretionary inputs, since they cannot be improved by the management.

Symbol $\epsilon > 0$ implies that the slack variables are to be handled at a later stage, since ϵ is a number so small, usually in the class of 10^6, that the contribution of the slacks to the objective function when compared to θ's is null, and does not affect its optimal value. Lastly, the slacks $s_i{}^-, i \in ND$ are omitted from the objective function entirely, since what is beyond the management's control cannot be used to evaluate a DMU's managerial decisions.

The improved activities of DMU o when the optimal solution is $(\theta^\star, \underline{s^-}^\star, \underline{s^+}^\star, \underline{\lambda_k}^\star)$ would then be:

$$
\begin{aligned}
\hat{x}_{io} &= \theta^\star x_{io} - s_i{}^-, \; i \in D \\
\hat{x}_{io} &= x_{io} - s_i{}^-, \; i \in ND \\
\hat{y}_{jo} &= y_{jo} + s_j{}^+, \; j = 1, \ldots, m.
\end{aligned}
$$

The model's dual form is as follows.

$$
\begin{aligned}
\max \sum_{j=1}^{m} u_j y_{jo} - \sum_{i \in ND} v_i x_{io} & \\
s.t. \ \sum_{j=1}^{m} u_j y_{jk} - \sum_{i \in ND} v_i x_{ik} - & \sum_{i \in D} v_i x_{ik} \leq 0, \ k = i, \ldots, K \\
& \sum_{i \in D} v_i x_{io} = 1 \\
& \quad u_i \geq 0, \ i \in ND \\
& \quad u_i \geq \epsilon, \ i \in D \\
& \quad v_j \geq \epsilon, \ j = 1, \ldots, m
\end{aligned}
$$

So only the non-discretionary inputs enter the objective function. Also, the constrains become such that we allow the multipliers of the non-discretionary inputs to be zero, since their contribution to the DMU's evaluation is not necessary if it doesn't positively impact its evaluation. If, for any $i \in ND$, $u_i^\star > 0$ then the efficiency score is reduced by the multiplier x_{io}.

Example 5.1

An example to such a non-discretionary input, is the population of the location around the store. For example, a store at a big city can be expected to have a bigger clientele than a store at a remote village. So, say that the population of the $1000m^2$ around the store (in the hundreds) is also added as a variable to the store's efficiency.

To illustrate the difference it can make in each store's evaluation, we will use the preexisting premises of Example 3.1, while adding as an extra, non-discretionary, input the population around each store (Table 5.1).

Locations	A	B	C	D	E	F	G	H	I	J
Local Population	8	2	9	3	10	4	9	2	12	5
Money provided	10	3	5	9	7	6	8	2	4	8
Sales	17	9	29	12	17	29	30	8	10	25
Customer satisfaction	10	15	10	10	1	6	4	10	9	9

Table 5.1: Single Input and Two Outputs Case with Local Population

The model would then become:

$$
\begin{aligned}
< C > \quad & \min \theta \\
s.t.\ & 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \geq 29 \\
& 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 10 \\
& 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 5\theta \\
& 8\lambda_1 + 2\lambda_2 + 9\lambda_3 + 3\lambda_4 + 10\lambda_5 + 4\lambda_6 + 9\lambda_7 + 2\lambda_8 + 12\lambda_9 + 5\lambda_{10} \leq 9 \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},
\end{aligned}
$$

where $\theta^\star = 1$, and the second phase becomes:

$$
\begin{aligned}
< C > \quad & \max w = s_1^- + s_2^- + s_1^+ + s_2^+ \\
& s.t.\ s_1^+ = -29 + 17\lambda_1 + 9\lambda_2 + 29\lambda_3 + 12\lambda_4 + 17\lambda_5 + 29\lambda_6 + 30\lambda_7 + 8\lambda_8 + 10\lambda_9 + 25\lambda_{10} \\
& s_2^+ = -10 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\
& s_1^- = 5\theta^\star - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\
& s_2^- = 9 - 8\lambda_1 - 2\lambda_2 - 9\lambda_3 - 3\lambda_4 - 10\lambda_5 - 4\lambda_6 - 9\lambda_7 - 2\lambda_8 - 12\lambda_9 - 5\lambda_{10} \\
& \underline{s}^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0},
\end{aligned}
$$

Doing this for every store, we get the following results (5.2).

Locations	A	B	C	D	E	F	G	H	I	J
With Population	0.33	1	1	0.3	0.42	1	0.65	1	0.57	0.63
Without Population	0.33	1	1	0.29	0.42	0.83	0.65	1	0.55	0.54

Table 5.2: The CCR-efficiency With and Without Non-discretionary Input

It is evident in this example that all the efficiency scores are either unaffected by the non-discretionary variable (see A, B, C, E, G, and H), or improved (see D, F, I and J), since their bad input situations are taken into consideration, giving a more lenient evaluation for their management style.

This model has its correspondent BCC model, in which the constraint $\sum_{k=1}^{K} \lambda_k = 1$, is added, with the rest staying the same.

5.3 Categorical Variables

The usual CCR and BCC models all fail to take into account what happens if one or more inputs are 0-1 variables, where the variable indicates simply if a DMU has a certain feature or not, or variables that indicate in which of a finite number of categories it is in.

The latter can also be used when the group of DMUs is not homogeneous, by comparing each DMU only with its homogeneous subgroup and those in less favorable situations than them. That can be done by distinguishing them into homogeneous categories such as “small”, “medium”,“big” or as “none”, “low”, “average” and “high”, where “high” denotes the most favorable situation.

In practice this can be done by defining a new, categorical variable d^p_{rk}, $r = 1, \ldots, s-1$, $p = 1, \ldots, L$ for the s categories that the L categorical characteristics have. This variable, d^p_{rk}, for a given characteristic p takes the value of 0 if the DMU k is on the categories $\{1, \ldots, (r-1)\}$ or 1 if it belongs on one of the others ($\{r, \ldots, s\}$).

So, if $\underline{d^p_k} = (0, 0, \ldots, 0)$ that means that the DMU k is in the first category of $p-th$ characteristic. If $\underline{d^p_k} = (1, 0, \ldots, 0)$, it means that its on the second category. If $\underline{d^p_k} = (1, 1, 0, \ldots, 0)$, it means its on the third, and so on, til $\underline{d^p_k} = (1, \ldots, 1)$ means that it belongs to the last category.

To encompass the model to include these variables, we add a constraint that ensures that for each DMU the peer group, which consists of the DMUs with $\lambda_k > 0$, does not include DMUs that are in a bigger category. To do that we must force the λ_k of each larger DMU to be zero. With this goal in mind, we add the following constraints.

$$\sum_{k=1}^{K} d^p_{rk}\lambda_k \leq d^p_{ro},\ r = 1, \ldots, s-1,\ p = 1, \ldots, L. \tag{5.1}$$

This way, if, say the DMU o belongs in the third category, $d^p_{ro} = 0$ for $r = 3, \ldots, s-1$, and so, for every DMU that has $d^p_{rk} = 1$ for at least one $r = 3, \ldots, s-1$, meaning a “bigger” DMU, the above constraint forces λ_k to be zero, since all products are non-negative and d^p_{rk} is positive. This way, (5.1) excludes all the “bigger” DMUs from its peer group.

The complete model, then, takes the following form.

$$
\begin{aligned}
&\min \theta - \epsilon \left(\sum_{i=1}^{n} s_i^- + \sum_{j=1}^{m} s_j^+ \right) \\
&s.t.\ \theta x_{io} = \sum_{k=1}^{K} x_{ik}\lambda_k + s_i^-,\ \ i = 1, \dots, n \\
&\qquad d_{ro}^p \geq \sum_{k=1}^{K} d_{rk}^p \lambda_k,\ r = 1, \dots, s-1,\ p = 1, \dots, L \\
&\qquad y_{jo} = \sum_{k=1}^{K} y_{jk}\lambda_k - s_j^+,\ \ j = 1, \dots, m \\
&\qquad \theta \in \mathbb{R}, s_i^-, s_j^+, \lambda_k \geq 0,
\end{aligned}
$$

which, once again can be transformed to the corresponding BCC model by adding the constraint $\sum_{k=1}^{K} \lambda_k = 1$.

Example 5.2

We will illustrate the different way non-discretionary inputs are handled using the same example as above, but without using the exact population as a variable. Instead, we will place each location in categories. In particular, locations with a local population of 1 through 5 hundred will be "small", locations between 6 and 10, "medium", and locations with bigger local population "large". That will be represented by $\underline{d} \in \{0,1\}^2$ where "small" will be $(0,0)$, "medium" will be $(1,0)$, and "large" will be $(1,1)$.

So our data now takes the form (Table 5.3):

Locations	A	B	C	D	E	F	G	H	I	J
Local Population	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,1)$	$(0,0)$
Money provided	10	3	5	9	7	6	8	2	4	8
Sales	17	9	29	12	17	29	30	8	10	25
Satisfaction	10	15	10	10	1	6	4	10	9	9

Table 5.3: Single Input and Two Outputs Case with Local Population

Using the model, as shown above but separated to two phases, for location C, which is a "medium" location, we have:

$$
\begin{array}{ll}
<C> & \min \theta \\
s.t. & \lambda_1 + \lambda_3 + \lambda_5 + \lambda_7 + \lambda_9 \leq 1 \\
& \lambda_9 \leq 0 \\
& 10\lambda_1 + 3\lambda_2 + 5\lambda_3 + 9\lambda_4 + 7\lambda_5 + 6\lambda_6 + 8\lambda_7 + 2\lambda_8 + 4\lambda_9 + 8\lambda_{10} \leq 5\theta \\
& 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \geq 10 \\
& 12\lambda_1 + 2\lambda_2 + 9\lambda_3 + 12\lambda_4 + 10\lambda_5 + 4\lambda_6 + 5\lambda_7 + 3\lambda_8 + 13\lambda_9 + 12\lambda_{10} \leq 9 \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0},
\end{array}
$$

which forces the location I out of C's peer group. The value of this problem is $\theta^\star = 1$, and the second phase becomes:

$$
\begin{array}{ll}
<C> & \max w = s^- + s_1^+ + s_2^+ \\
& s.t.\ 1 \geq \lambda_1 + \lambda_3 + \lambda_5 + \lambda_7 + \lambda_9 \\
& 0 \geq \lambda_9 \\
& s^- = 5\theta^\star - 10\lambda_1 - 3\lambda_2 - 5\lambda_3 - 9\lambda_4 - 7\lambda_5 - 6\lambda_6 - 8\lambda_7 - 2\lambda_8 - 4\lambda_9 - 8\lambda_{10} \\
& s_1^+ = -10 + 10\lambda_1 + 15\lambda_2 + 10\lambda_3 + 10\lambda_4 + \lambda_5 + 6\lambda_6 + 4\lambda_7 + 10\lambda_8 + 9\lambda_9 + 9\lambda_{10} \\
& s_2^+ = -9 + 12\lambda_1 + 2\lambda_2 + 9\lambda_3 + 12\lambda_4 + 10\lambda_5 + 4\lambda_6 + 5\lambda_7 + 3\lambda_8 + 13\lambda_9 + 12\lambda_{10} \\
& \underline{s}^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0},
\end{array}
$$

where we have the final result.

The efficiency of each store thus becomes:

Locations	A	B	C	D	E	F	G	H	I	J
CCR-efficiencies	0.33	1	1	0.32	0.47	1	1	1	0.57	1

Table 5.4: The CCR-efficiency With the Categorical Input

In this way, we can see that this model allows F and G and J to become efficient as well. The efficiency score here is still the same or improved, when compared to the original problem, and even when compared to the simple non-discretionary CCR problem above.

5.4 Conclusion

In this chapter we analysed special models for when the conditions for a simple LP CCR or BCC model are not quite met. In particular, in the case of inputs that are not under managerial control, such as the weather or the population of a location we cannot assume that they can be improved by changing managerial models, and so it is nonsensical to judge a DMU for them, or to leave them out of the evaluation entirely. So, in this model, we excluded from the objective function the slacks that are caused by these non-discretionary variables, and disallowed any radial improvement to these inputs. Thus inputs that are not under the control of management can be taken under consideration while taking into account that they cannot be changed.

Next, we analyzed ways to integrate categorical variables and using that, how to take a non-homogeneous system of DMUs and judge each DMU only with its similar or those less in less favorable categories. That was done under the assumption that the categorical variables are non-discretionary.

Part II

An Example of DEA in Queues

Chapter 6

An Application in Queues

6.1 Problem Description

In this chapter we will attempt to use the DEA methods to evaluate the performance of Markov queuing systems and to determine their efficiency in regards to their average performance. The idea is to evaluate Markov queues with k servers and capacity s, where customers join the system if there is capacity for them, otherwise they are blocked and do not receive service.

To frame this in more realistic terms, say there is a businesswoman that can spare 7.2 thousand dollars per month for her store. Due to constrictions that came about in this epidemic, there can only be one customer for each $10m^2$. The cost every $10m^2$ is 200$ per month and the monthly salary for a server is 1000$. In this example we will explore the possible options she has with the money she is willing to put into it. In particular, we will explore the store's efficiency with each possible choice of capacity and servers, as well as three different ways in which she may use them, in regards to the customer satisfaction (judged by the average waiting and sojourn time) and the percentage of customers served in long-term.

For each possible choice in the number of servers and the capacity, she may choose to create a system that has a single queue and k parallel servers, a system that has one queue but all the servers are working on a single customer together, thus multiplying the service rate by k, or a system that has k parallel queues that are served by a single server which divides the capacity and the arrival rate of each queue by k. The expected arrival rate of customers is 1 per 10 minutes and the service rate of each server is 1 service per 50 minutes.

To begin with, we first need to calculate the average percentage of customers lost and the average waiting and sojourn times. Then, we apply DEA to evaluate the different systems that can be constructed.

6.2 Computation of inputs and outputs

As a general rule, the possible systems she may use fall under the general category of M|M|k|s, where the arrival rate is λ and the service rate is μ. To calculate the needed quantities, we follow the steps below.

The first thing we need, then, is the stationary distribution of the number of customers in the system for M|M|k|s, which exists and is unique since the queue is finite.

Step 1: We first compute the stationary distribution of the number of customers in the system for M|M|k|s, which coincides with the limiting distribution since it is irreducible. Let $\rho = \frac{\lambda}{\mu}$ and Q_s the number of customers in the system in steady state. Then

$$\pi_n = \mathbf{P}(Q_s = n) = \begin{cases} B\frac{\rho^n}{n!} & , 0 \le n \le k \\ B\frac{\rho^n}{k!}\frac{1}{k^{n-k}} & , k+1 \le n \le s \, , \end{cases}$$

where

$$B^{-1} = \sum_{n=0}^{k} \frac{\rho^n}{n!} + \sum_{m=k+1}^{s} \frac{\rho^m}{k!}\frac{1}{k^{m-k}}.$$

Step 2: We compute the average percentage of customers served, N.

$$N_{ks} = 1 - \pi_s.$$

Step 3: We compute the average Waiting Time.

Since the average queue length is $Q_q = \sum_{n=1}^{s-k} n\pi_{n+k}$, and the effective arrival rate of customers that enter the system is $\lambda^\star = \lambda\,(1-\pi_s)$, we can use Little's law to calculate the average waiting time of a customer that enters the system by:

$$W_{ks} = \frac{Q_q}{\lambda^\star} = \frac{\sum_{n=1}^{s-k} n\pi_{n+k}}{\lambda\,(1-\pi_s)}.$$

Step 4: The average Sojourn Time are the average Waiting Time plus the average service time.

$$S_{ks} = W_{ks} + \frac{1}{\mu}.$$

Step 5: Lastly, we calculate all the quantities for the first setting M|M|k|s (arrivals λ, service μ), henceforth called *Parallel Servers*, the second one, M|M|1|s (arrivals λ, service $k\mu$), called *Cooperating Servers*, and M|M|1|$\frac{s}{k}$ (arrivals $\frac{\lambda}{k}$, service μ), called *Parallel Queues*, for $k = 2, \ldots, 6$ and $s = 36 - 5 \times k$. Doing so for $\lambda = 1$ and $\mu = \frac{1}{5}$, we get the following (Tables 6.1, 6.2, 6.3), using the following **R** code for mu=1/5 and l=1:

```
MMks=function(l,mu,k,s){
  r=l/mu
  m=(k+1):s
  n=0:k
  #------Forming B--------------------
  rk=r^n/factorial(n)
  rs=c()
  if (k<s){
    rs=r^m/(factorial(k)*k^(m-k))
  }
  rr=c(rk,rs)
  B=sum(rr)
  B=1/B
  #-----Stationary Distribution--------
  p=B*rr
  #-----Percentage of customers served-
  ps=1-p[s+1]
  #-----Average Queue------------------
  lstar=l*ps
  EQ=sum(p[m]*(m-k))
  #-----Average Waiting Time-----------
  EW=EQ/lstar
  #-----Average Sojourn Time-----------
  ES=EW+1/mu
  EW=1/EW
  ES=1/ES
  re=list(per=ps,REW=EW,RES=ES)
  return(re)
}
```

```
l=1; mu=1/5
x=c(); all1=c(); c=36
for (k in 2:6){
  s=c-5*k
  # 1 queue k customers served, 1 queue 1 customer served,
  # k queues k customers served
  all2=cbind(MMks(l,mu,k,s),MMks(l,mu*k,1,s),MMks(l/k,mu,1,floor(s/k)))
```

```
  all1=cbind(all1,all2)
  x=cbind(x,c(k,s),c(k,s),c(k,s))#
}
y=all1
```

Table 6.1: Parallel Servers

Names	A_1	A_2	A_3	A_4	A_5
k	2	3	4	5	6
s	26	21	16	11	6
N_{ks}	0.40	0.60	0.79	0.89	0.81
W_{ks}	23.33	16.5	8.64	2.47	0.24
S_{ks}	28.33	21.5	13.64	7.47	5.24

Table 6.2: Cooperating Servers

Names	B_1	B_2	B_3	B_4	B_5
k	2	3	4	5	6
s	26	21	16	11	6
N_{ks}	0.40	0.60	0.80	0.92	0.92
W_{ks}	24.33	18.5	11.46	5	1.98
S_{ks}	26.83	20.17	12.71	6	2.81

Table 6.3: Parallel Queues

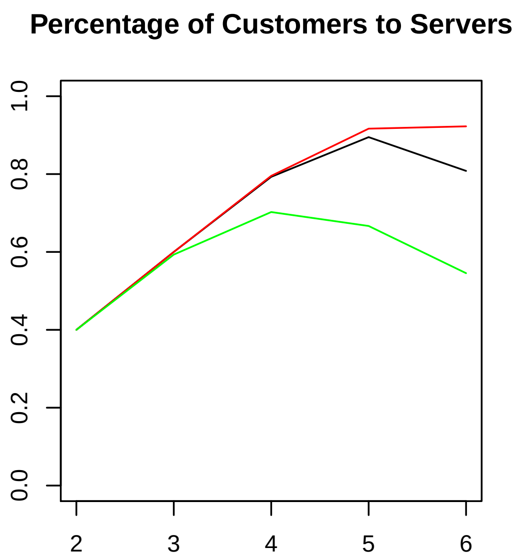
Names	C_1	C_2	C_3	C_4	C_5
k	2	3	4	5	6
s	26	21	16	11	6
N_{ks}	0.40	0.59	0.70	0.67	0.54
W_{ks}	22.67	14.1	7.1	2.5	0
S_{ks}	27.67	19.1	12.1	7.5	5

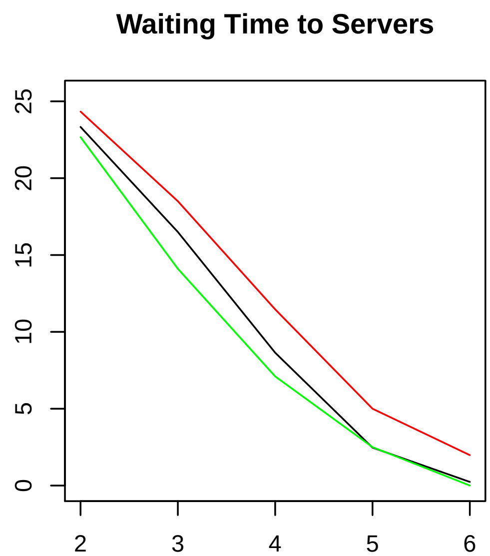
We remove C_5 from the data base, as it leads to an infinite reverse waiting time, and will simply consider it an efficient queue. In general, we can observe that, though not actually increasing, the percentage of customers served has an upwards tendency until it reaches its highest point and then begins descending. The change in monotonicity is probably brought on by the lessening of the capacity, as more and more customers that arrive are lost. Here is a plot of the percentage of customers served to the number of servers in each queue, to make envision easier. The black line represents Parallel

Servers, the red Cooperating Servers, and the green one Parallel Queues. The red line, that of Cooperating Servers, is above the rest at all times, and the green one is always below the rest (or equivalent), for each different number of servers.

As for the waiting time, it is almost always decreasing. And in that decreasing state, the Cooperating Servers make for the most waiting time in each different Server-Capacity choice, and the Parallel Queues the least. The increased incline in the line before is probably the reason that the percentage of customers served in the Parallel Queues starts getting smaller at C_4. In general, though, the red is never the best option. We illustrate it all once again in a plot, using the coloured lines as described above.

Doing the same for the sojourn time, we can observe that the Cooperating Servers queues and the Parallel Queues ones alternate for the least sojourn time, while the Parallel Servers queues are consistently not the best in that regard.





It can be observed that while the queue with cooperating servers has a greater (or equal) percentage of customers served, for each different server-capacity arrangement, its average waiting times are generally (with the exception of C_5) larger. On the other hand, its sojourn times are also pretty good, as with the exception of C_2 and C_3 all other corresponding sojourn times are larger. The parallel queues seems to lose a greater percentage of customers than the rest.

6.3 CCR models

To start with, in a DEA model, a DMU is considered to be more efficient as the outputs increase, so to apply those methods we need first to alter the two last outputs, since the smaller they are, the better the stores efficiency is. One way to do that is to invert them. Another is to use their distance from their maximum.

By inverting the average times, we can then create the 2-Phase input-oriented CCR model which will give us their CCR-efficiency as well as their slacks and improved activities. The input-oriented model is chosen in this case because in this example we can handle inputs easier. The CCR model must be performed for all queues listed in the tables (6.1,6.2,6.3).

As an example, we will present analytically the model for the queue A_1. The rest will be calculated through a program in **R** and presented here in the form of a table.

$$\begin{aligned}
< A_1 > \quad & \min\ \theta \\
s.t.\ & 0.4\lambda_1 + 0.6\lambda_2 + 0.79\lambda_3 + 0.89\lambda_4 + 0.86\lambda_5 + 0.4\lambda_6 + 0.6\lambda_7 + 0.8\lambda_8 + \\
& \qquad 0.92\lambda_9 + 0.92\lambda_{10} + 0.4\lambda_{11} + 0.59\lambda_{12} + 0.7\lambda_{13} + 0.67\lambda_{14} \geq 0.4 \\
& 0.04\lambda_1 + 0.06\lambda_2 + 0.12\lambda_3 + 0.41\lambda_4 + 4.21\lambda_5 + 0.04\lambda_6 + 0.05\lambda_7 + 0.09\lambda_8 + \\
& \qquad 0.2\lambda_9 + 0.51\lambda_{10} + 0.04\lambda_{11} + 0.07\lambda_{12} + 0.14\lambda_{13} + 0.4\lambda_{14} \geq 0.04 \\
& 0.04\lambda_1 + 0.05\lambda_2 + 0.07\lambda_3 + 0.13\lambda_4 + 0.19\lambda_5 + 0.04\lambda_6 + 0.05\lambda_7 + 0.08\lambda_8 + \\
& \qquad 0.17\lambda_9 + 0.36\lambda_{10} + 0.04\lambda_{11} + 0.05\lambda_{12} + 0.08\lambda_{13} + 0.13\lambda_{14} \geq 0.04 \\
& 2\lambda_1 + 3\lambda_2 + 4\lambda_3 + 5\lambda_4 + 6\lambda_5 + 2\lambda_6 + 3\lambda_7 + 4\lambda_8 + \\
& \qquad 5\lambda_9 + 6\lambda_{10} + 2\lambda_{11} + 3\lambda_{12} + 4\lambda_{13} + 5\lambda_{14} + 6\lambda_{15} \leq 2\theta \\
& 26\lambda_1 + 21\lambda_2 + 16\lambda_3 + 11\lambda_4 + 6\lambda_5 + 26\lambda_6 + 21\lambda_7 + 16\lambda_8 + \\
& \qquad 11\lambda_9 + 6\lambda_{10} + 26\lambda_{11} + 21\lambda_{12} + 16\lambda_{13} + 11\lambda_{14} + 6\lambda_{15} \leq 26\theta \\
& \theta \in \mathbb{R},\ \underline{\lambda} \geq \underline{0}.
\end{aligned}$$

Solving this problem we obtain $\theta^\star = 1$.

The second phase, then, is

$$
\begin{aligned}
< A_1 > \quad \max\ w =& s_1^- + s_2^- + s_1^+ + s_2^+ + s_3^+ \\
s.t.\ \ s_1^+ =& -0.4 + 0.4\lambda_1 + 0.6\lambda_2 + 0.79\lambda_3 + 0.89\lambda_4 + 0.86\lambda_5 + 0.4\lambda_6 + 0.6\lambda_7 + 0.8\lambda_8 + \\
& 0.92\lambda_9 + 0.92\lambda_{10} + 0.4\lambda_{11} + 0.59\lambda_{12} + 0.7\lambda_{13} + 0.67\lambda_{14} \\
s_2^+ =& -0.04 + 0.04\lambda_1 + 0.06\lambda_2 + 0.12\lambda_3 + 0.41\lambda_4 + 4.21\lambda_5 + 0.04\lambda_6 + 0.05\lambda_7 + 0.09\lambda_8 + \\
& 0.2\lambda_9 + 0.51\lambda_{10} + 0.04\lambda_{11} + 0.07\lambda_{12} + 0.14\lambda_{13} + 0.4\lambda_{14} + 0.2\lambda_{15} \\
s_3^+ =& -0.04 + 0.04\lambda_1 + 0.05\lambda_2 + 0.07\lambda_3 + 0.13\lambda_4 + 0.19\lambda_5 + 0.04\lambda_6 + 0.05\lambda_7 + 0.08\lambda_8 + \\
& 0.17\lambda_9 + 0.36\lambda_{10} + 0.04\lambda_{11} + 0.05\lambda_{12} + 0.08\lambda_{13} + 0.13\lambda_{14} \\
s_1^- =& 2(\theta^\star) - 2\lambda_1 - 3\lambda_2 - 4\lambda_3 - 5\lambda_4 - 6\lambda_5 - 2\lambda_6 - 3\lambda_7 - 4\lambda_8 - \\
& 5\lambda_9 + 6\lambda_{10} + -2\lambda_{11} - 3\lambda_{12} - 4\lambda_{13} - 5\lambda_{14} \\
s_2^- =& 26(\theta^\star) - 26\lambda_1 - 21\lambda_2 - 16\lambda_3 - 11\lambda_4 - 6\lambda_5 - 26\lambda_6 - 21\lambda_7 - 16\lambda_8 - \\
& 11\lambda_9 - 6\lambda_{10} - 26\lambda_{11} - 21\lambda_{12} - 16\lambda_{13} - 11\lambda_{14} \\
& \underline{s}^- \geq 0,\ \underline{s}^+ \geq \underline{0},\ \underline{\lambda} \geq \underline{0}.
\end{aligned}
$$

```
CCRdual=function(o,input, output){
  if (is.vector(input)) input=t(matrix(input))
  if (is.vector(output)) output=t(matrix(output))
  x=input; y=output
  K=length(x[1,])
  m=length(y[,1])
  n=length(x[,1])
  c=c(1,-1,rep(0,K+n+m))
  A=cbind(x[,o],-1*x[,o],-1*x)
  a1=cbind(rep(0,m),rep(0,m),y)
  A=rbind(A,a1)
  A=cbind(A,-diag(n+m))
  b=c(rep(0,n),y[,o])
  sol=simplex(a = c, A3 = A, b3 = b)
  l=sol$soln[3:(K+2)]
```

```
  #---optimal solution
  theta=sol$soln[1]-sol$soln[2]
  #---efficient frontier
  so=sol$value
  c1=c(rep(1,n+m),rep(0,K))
  b1=c(theta*x[,o],y[,o])
  A1=diag(c(rep(1,n),rep(-1,m)))
  a=rbind(x,y)
  A1=cbind(A1,a)
  dir=rep("==",length(b1))
  sol2=Rglpk_solve_LP(c1,A1,dir,b1, max=TRUE)
  s_m=sol2$solution[1:n]
  s_p=sol2$solution[(n+1):(n+m)]
  l2=sol2$solution[(n+m+1):(n+m+K)]
  CCR=list(theta=theta,lambda=l,la2=l2,s_m=s_m,s_p=s_p,sol=so)
  return(CCR)
}
```

The results we get by implementing the corresponding model for each queue are presented in the following Tables. CCR efficiency scores are presented in Table 6.4:

Table 6.4: Ratio Efficiency

Names	1	2	3	4	5
Parallel Servers (A)	1	1	1	0.99	1
Cooperating Servers (B)	1	1	1	1	1
Parallel Queues (C)	1	0.992	0.904	0.756	-

The slacks are presented in the following Tables 6.5,6.6,6.7:

Table 6.5: Parallel Servers

Names	A_1	A_2	A_3	A_4	A_5
s_1^-	0	0	0	0	0
s_2^-	0	0	0	0	0
s_1^+	0	0	0	0	0
s_2^+	0	0	0	0	0
s_3^+	0	0	0	0.028	0

Table 6.6: Cooperating Servers

Names	B_1	B_2	B_3	B_4	B_5
s_1^-	0	0	0	0	0
s_2^-	0	0	0	0	0
s_1^+	0	0	0	0	0
s_2^+	0	0	0	0	0
s_3^+	0	0	0	0	0

Table 6.7: Parallel Queues

Names	C_1	C_2	C_3	C_4
s_1^-	0	0	0	0
s_2^-	0	0	0.977	0
s_1^+	0	0	0	0
s_2^+	0	0	0	0
s_3^+	0	0	0	0

As for the λ, it is presented in the following table (6.8)

Table 6.8: λ

Names	A_1	A_2	A_3	A_4	A_5	B_1	B_2	B_3	B_4	B_5	C_1	C_2	C_3	C_4
A_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
A_2	0	1	0	0	0	0	0	0	0	0	0	0.43	0	0
A_3	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A_5	0	0	0	0.05	1	0	0	0	0	0	0	0	0.01	0.06
B_1	0	0	0	0	0	1	0	0	0	0	0	0	0.06	0
B_2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
B_3	0	0	0	0.05	0	0	0	1	0	0	0	0.27	0.79	0.17
B_4	0	0	0	0.88	0	0	0	0	1	0	0	0	0	0.42
B_5	0	0	0	0	0	0	0	0	0	1	0	0	0.05	0.11
C_1	0	0	0	0	0	0	0	0	0	0	1	0.28	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0

We can observe in Table (6.4) that when comparing the different systems, a queue with Cooperating Servers is always at least as efficient as the rest. Also, the system of having Parallel Queues is always the least efficient. The former is brought upon due to having better admission rate than the rest. It does not take into account the average waiting and sojourn times, that can on occasion be worse. That can be demonstrated by the fact that if you solve the multiplier problem, the multipliers for them in the cooperating servers' systems are all zero, as can be seen in the following tables that present all the multipliers (6.9,6.10,6.11). It is also evident through the slacks that the mix inefficiencies present in the DMUs are caused by the second and third output factors.

The weights calculated from the multiplier LP, are the following

Table 6.9: Parallel Servers

Names	A_1	A_2	A_3	A_4	A_5
v_1	0.5	0.33	0.24	0.18	0.17
v_2	0	0	0	0.01	0
u_1	2.5	1.67	1.25	1.08	0
u_2	0	0	0.11	0.06	0.24
u_3	0	0	0	0	0

Table 6.10: Cooperating Servers

Names	B_1	B_2	B_3	B_4	B_5
v_1	0.5	0.33	0.24	0.18	0.17
v_2	0	0	0	0.01	0
u_1	2.5	1.67	1.26	1.09	1.08
u_2	0	0	0	0	0
u_3	0	0	0	0	0

Table 6.11: Parallel Queues

Names	C_1	C_2	C_3	C_4
v_1	0.5	0.33	0.25	0.2
v_2	0	0	0	0
u_1	2.5	1.63	1.13	0.91
u_2	0	0.15	0.09	0.07
u_3	0	0.27	1.15	0.91

Also, from the table that contains λ we can easily deduce each system's peer group. They are the systems that have a positive λ in the corresponding column. They are presenting in Table 6.12.

Table 6.12: Peer Group

A	Peer Group
1	A_1
2	A_2
3	A_3
4	A_5, B_3, B_4
5	A_5

B	Peer Group
1	B_1
2	B_2
3	B_3
4	B_4
5	B_5

C	Peer Group
1	C_1
2	A_2, B_3, C_1
3	A_5, B_1, B_3, B_5
4	A_5, B_3, B_4, B_5
5	-

Should the first factor, the percentage of customers served, be restricted to hold less weight in the estimation of their efficiency than the waiting time, the difference in their efficiency is stark. (Table 6.13)

Table 6.13: CCR Efficiency

Names	1	2	3	4	5
Parallel Servers (A)	0.396	0.365	0.409	0.551	1
Cooperating Servers (B)	0.410	0.378	0.424	0.621	1
Parallel Queues (C)	0.403	0.393	0.432	0.519	-

That means that the first factor, the percentage of customers served, has a very large effect on the efficiency of each queue. Forcing it to have less effect on the efficiency than the waiting time, results in far less relative efficiency, and only two DMUs are actually CCR-efficient. The ones with 6 servers and 6 capacity, that have Cooperating or Parallel Servers are the only ones with a CCR-efficiency of 1. Also most of the queue with Cooperating Servers is now inefficient, something that can be anticipated as, in the plots that where shown above, we observed that the red line was always above the rest when it comes to the waiting times. That is a drawback to the DEA method of evaluation, since it is almost too generous in the selection of weights. A DMU will, if it is favorable to it, completely disregard the factors that are not as good as the rest's, which can lead to it being falsely presented as more efficient than others.

There are ways, as shown above, to make sure that their evaluation puts more weight to the other factors, by constraining their multipliers to be more than the problematic one's, but that cannot be done simultaneously for all factors, and thus is once again subjective. Another way would be to force all the multipliers to be larger than a certain small threshold, so as to avoid leaving entire factors out of the DMU's evaluation.

We have created the following **R** function for this.

```
CCRkl=function(i,input, output,k,l){
  #----We need the data to be in matrix form
  if (is.vector(input)){
    input=t(matrix(input))
  }
  if (is.vector(output)){
    output=t(matrix(output))
  }
  x=input; y=output
  n=length(x[1,])
  s=length(y[,1])
  m=length(x[,1])
  #---Modeling the problem
  A=cbind(t(y),t(-x))
  d=t(c(rep(0,s),x[,i]))
  A=rbind(A,d,-d)
  kk=rep(0,s+m)
  kk[k]=-1;kk[l]=1
  A=rbind(A,kk)

  c=c(y[,i], rep(0,m))

  b=c(rep(0,n),1,-1,0)

  sol=simplex(a=c,A1=A,b1=b,maxi=T)
  #---optimal weights
  v=as.vector(sol$soln[(s+1):(s+m)])
  u=as.vector(sol$soln[1:s])
  #---optimal solution
  theta=as.vector(sol$value)
  #---efficient frontier
  ef=c()
  for(j in 1:n){
    if(abs(v%*%x[,j]-u%*%y[,j])<10^(-10)){
      ef=c(ef,j)
    }
  }
```

```
  CCR=list(u=u,v=v,theta=theta,ef.frontier=ef,dual=sol$solution_dual)
  return(CCR)
}
```

Here, for example, we can demand that all multipliers be larger than 0.01. That would lead to the following CCR-efficiency Table (6.14).

Table 6.14: CCR Efficiency

Names	1	2	3	4	5
Parallel Servers (A)	0.825	0.915	0.999	0.989	1
Cooperating Servers (B)	0.825	0.915	1	1	1
Parallel Queues (C)	0.825	0.905	0.888	0.748	-

6.4 The improved activities

Back to the original problem, to conclude, the improved activities of the queues can be calculated by using formulae (3.11),(3.12),

$$\underline{\hat{x}_o} = \theta^\star \underline{x_o} - \underline{s}^{-\star} = \underline{\lambda}' \underline{x_o} \leq \underline{x_o},$$
$$\underline{\hat{y}_o} = \underline{y_o} + \underline{s}^{+\star} = \underline{\lambda}' \underline{y_o} \geq \underline{y_o}.$$

while keeping in mind that the inputs are integers, and rounding them up, as well as inverting the average times, to get the following (6.15,6.16,6.17).

Table 6.15: Parallel Servers

Names	A_1	A_2	A_3	A_4	A_5
k	2	3	4	5	6
s	26	21	16	11	6
N_{ks}	0.4	0.6	0.79	0.89	0.81
W_{ks}	23.33	16.5	8.64	2.47	0.24
S_{ks}	28.33	21.5	13.64	6.19	5.24

Table 6.16: Cooperating Servers

Names	B_1	B_2	B_3	B_4	B_5
k	2	3	4	5	6
s	26	21	16	11	6
N_{ks}	0.40	0.60	0.80	0.92	0.92
W_{ks}	24.33	18.5	11.46	5	1.98
S_{ks}	26.83	20.17	12.71	6	2.81

Table 6.17: Parallel Queues

Names	C_1	C_2	C_3	C_4	C_5
k	2	3	4	4	4
s	26	21	14	8	4
N_{ks}	0.4	0.59	0.7	0.67	0.55
W_{ks}	22.67	14.1	7.1	2.5	3.35
S_{ks}	27.67	19.1	12.1	7.5	4.76

That is to say, that if the number of servers and capacity is improved as shown above then the systems will be more efficient. The outputs here are not the expected results of using these particular queuing systems. The best way to interpret the resulting improved activities is that they are a mix of the strategies of the queues in their peer group according to their λ. That is to say that, for example, C_2 could be improved by adopting the system A_2 about 443% of the time, B_3 about 28% of the time, and C_1 28% of the time, while C_5 can be improved by implementing the system B_5.

Another way to find the best queuing systems could to first find the most efficient in each queue with the same system. That is to say, we solve the CCR models for all queues with Parallel Servers, with Cooperating Servers and with Parallel Queues, and then, if we need to find the most efficient one, we compare them again. The problem then, and the reason we don't use that method here, is that due to the smaller number of queues in the model, most systems end up being CCR-efficient, making the comparison a moot point.

In all, the CCR-inefficient queues have reference sets that include A_2, A_5, B_1, B_3, B_4, B_5, C_1. So these are the queues that force the inefficient queues to have a CCR-efficiency less than one. Also, since B_3 shows up in almost all the reference sets, it can be considered the representative Unit or the global leader.

6.5 The BCC model

Of course, the CCR model is not the only model we can use to evaluate them. Another possible option is to use the BCC model. Unfortunately, that leads to even softer and more inconclusive results, since it is a softer approach to evaluation of efficiency, as it judges the DMUs with far greater leniency.

An **R** function for calculating the relevant measures is shown below.

```
BCCm=function(i,input, output){
  #----We need the data to be in matrix form
  if (is.vector(input)){
    input=t(matrix(input))
  }
  if (is.vector(output)){
    output=t(matrix(output))
  }
  x=input; y=output
  n=length(x[1,])
  s=length(y[,1])
  m=length(x[,1])
  #---Modeling the problem
  A=cbind(t(y),t(-x))
  d=t(c(rep(0,s),x[,i]))
  A=rbind(A,d)
  A=cbind(A,c(rep(-1,n),0),c(rep(1,n),0))

  c=c(y[,i], rep(0,m),-1,1)

  b=c(rep(0,n),1)

  dir=c(rep("<=",n),"==")

  sol=Rglpk_solve_LP(c,A,dir,b, max=TRUE)
  #---optiman weights
  v=sol$solution[(s+1):(s+m)]
  u=sol$solution[1:s]
  #---optimal solution
  theta=sol$optimum   #£
```

```
BCC=list(u=u,v=v,theta=theta)   #}{
return(BCC)}
```

In this particular case, as a matter of fact, all the DMUs' BCC-efficiency is 1, which accommodates no comparison between them whatsoever. It is essentially not a measure of efficiency that can be used in this case, at all.

6.6 Alternate Models

This example was established for a small business in a place with a small local population, leading to a small arrival rate ($\lambda = 1$). Seeing how efficient the store was, bigger businessmen thought to implement this strategy to a store in a bigger city, and also to see if it would be more efficient to do so. The bigger local population is represented in a queue by having a larger arrival rate. In this example, the arrival rate will be $\lambda = 6$. Using the same model as above, and simply increasing the number of DMUs compared, we get the following CCR-efficiencies (6.18).

Table 6.18: CCR Efficiency

Names	1	2	3	4	5	6	7	8	9	10
Parallel Servers (A)	1	1	1	0.99	0.96	0.6	0.48	0.44	0.48	0.48
Cooperating Servers (B)	1	1	1	1	1	0.76	0.69	0.69	0.77	1
Parallel Queues (C)	1	0.99	0.9	0.76	-	0.6	0.49	0.45	0.57	0.24

Then we can see that most businesses in the city are not CCR-efficiency when compared to the businesses in a less populated area. That is, presumably, due to the smaller amount of money per customer that is allotted in the second case. To maybe make the comparison a more fair one, we can add the difference between the two potential locations by adding it as a non-discretionary variable through the arrival rate, λ. Then their efficiency will be computed with the following **R** code.

```
CCR_nd=function(o,input,output,ndinput){
  if (is.vector(input)){
    input=t(matrix(input))
  }
  if (is.vector(output)){
    output=t(matrix(output))
  }
  if (is.vector(ndinput)){
    ndinput=t(matrix(ndinput))
  }
  x=input; y=output; d=ndinput
  n=length(x[1,])
  m=length(x[,1])
  s=length(y[,1])
  r=length(d[,1])
```

```
  # matrix A
  th=rbind(t(t(-1*x[,o])),t(t(rep(0,r))),t(t(rep(0,s))))
  la=rbind(x,d,y)
  A=cbind(th,la)
  # objective
  c=c(1,rep(0,n))
  # sol vector
  b=c(rep(0,m),d[,o],y[,o])
  # dir
  dir=c(rep("<=",m+r),rep(">=",s))
  # the problem
  sol=Rglpk_solve_LP(c,A,dir,b)
  theta=round(as.vector(sol$solution[1]),4)
  # phase 2
  # matrix A
  la=rbind(x,d,y)
  sm=rbind(diag(m),matrix(0,r,m),matrix(0,s,m))
  sp=rbind(matrix(0,m,s),matrix(0,r,s),diag(-1,s))
  A=cbind(la,sm,sp)
  # objective
  c=c(rep(0,n),rep(1,m+s))
  # sol vector
  b=c(theta*x[,o],d[,o],y[,o])
  # dir
  dir=c(rep("==",m),rep("<=",r),rep("==",s))
  # the problem
  sol=Rglpk_solve_LP(c,A,dir,b,max = TRUE)
  lambda=round(as.vector(sol$solution[1:n]),4)
  s_m=round(as.vector(sol$solution[(n+1):(n+m)]),4)
  s_p=round(as.vector(sol$solution[(n+m+1):(n+m+s)]),4)
  CCR=list(theta=theta,lambda=lambda,s_m=s_m,s_p=s_p)
  return(CCR)
}
```

The results are presented in the following table (Table 6.19).

Table 6.19: CCR Efficiency with Arrival Rates

Names	1	2	3	4	5	6	7	8	9	10
Parallel Servers (A)	1	1	1	0.98	1	0.6	0.48	0.44	0.48	0.48
Cooperating Servers (B)	1	1	1	1	1	0.76	0.69	0.69	0.77	1
Parallel Queues (C)	1	0.99	0.9	0.76	-	0.6	0.49	0.45	0.57	0.24

As we can see above, the addition of the arrival rates makes a very small difference in the DMUs' efficiency score. It only makes A_5 efficient. All the rest efficiency scores remain pretty much the same.

Another model we can use to include the difference in difficulty, is a model with a categorical variable, d. Here, d will be used to reflect the level of difficulty in opening a small store in a bigger city. So $d = 1$ in the less populated area, and $d = 0$ in the more populated area. This way, the businesses in less populated places do not enter the peer group of the city ones. To get the results we need, we will use the **R** code presented below.

```
BK=function(o,input,output,dinput){
  if (is.vector(input)) input=t(matrix(input))
  if (is.vector(output)) output=t(matrix(output))
  if (is.vector(dinput)) dinput=t(matrix(dinput))
  x=input; y=output; d=dinput
  K=length(x[1,])
  m=length(y[,1])
  n=length(x[,1])
  s=length(d[,1])
  c=c(1,-1,rep(0,K+n+m+s))
  A=cbind(x[,o],-1*x[,o],-1*x)
  a1=cbind(rep(0,m),rep(0,m),y)
  A=rbind(A,a1)
  A=cbind(A,-diag(n+m))
  a2=cbind(rep(0,s),rep(0,s),d,matrix(0,s,n+m))
  A=rbind(A,a2)
  A=cbind(A,rbind(matrix(0,n+m,s),diag(s)))
  A=rbind(A,c(rep(0,2),rep(1,K),rep(0,n+m+s)))
  #  A=cbind(A,diag(c(rep(1,n),rep(-1,m)),n+m))
  b=c(rep(0,n),y[,o],d[,o],1)
  dir=rep("==",length(b))
```

```
  sol=Rglpk_solve_LP(c,A,dir, b)
  l=as.vector(sol$solution[3:(K+2)])
  #---optimal solution
  theta=as.vector(sol$solution[1]-sol$solution[2])
  #---efficient frontier
  so=as.vector(sol$optimum)

  c1=c(rep(1,n+m+s),rep(0,K))
  b1=c(theta*x[,o],y[,o],d[,o])
  A1=diag(c(rep(1,n),rep(-1,m),rep(0,s)))
  a=rbind(x,y,d)
  A1=cbind(A1,a)
  dir=c(rep("==",n+m),rep("<=",s))
  sol2=Rglpk_solve_LP(c1,A1,dir,b1, max=TRUE)
  s_m=as.vector(sol2$solution[1:n])
  s_p=as.vector(sol2$solution[(n+1):(n+m)])
  BK=list(theta=theta,lambda=l,s_m=s_m,s_p=s_p,sol=so)
  return(BK)
}
```

Table 6.20: CCR Efficiency with Categorical Variable

Names	1	2	3	4	5	6	7	8	9	10
Parallel Servers (A)	1	1	1	0.98	1	1	1	1	1	1
Cooperating Servers (B)	1	1	1	1	1	1	1	1	1	1
Parallel Queues (C)	1	0.99	0.9	0.76	-	1	1	1	1	0.83

This model does make more of the city businesses CCR-efficient, since only the city businesses can enter their peer group. In fact, it makes almost all the systems efficient. So, if you effectively exclude the small, little town, businesses, then they all, apart from C_{10}, can be justified in calling themselves CCR-efficient. But that is only by excluding the smaller businesses from their evaluation.

The justification for assuming that it is more difficult to make it in a bigger city, is largely substantiated by the less amount of money per customer in the city business. To test that assumption, say that the city businesses had more money to use per customer, than the little town one. Say, for example, that they had 48 thousand dollars to spend on a store. Then, the money spent per customer would be higher than the other one.

To have a more general result, of course, we can't not include in the efficiency score the difference in size, nor the difference in the customer base. So to take those differences into account, we will add a categorical variable, d, with $d_k = 0$ for the more difficult situation, the one where less funds are allocated per customer, which is still the business in the less populated place, and $d_k = 1$ for the city business.

Of course, having more money, it doesn't make sense to spend it all on capacity, so in this case, we will have queues with $k = 18, 21, 24, 27, 30$ servers, instead of having between 2 and 5, and the capacity will be computed by $s = 240 - 5k$.

Table 6.21: CCR Efficiency with Categorical Variable

Names	1	2	3	4	5	6	7	8	9	10
Parallel Servers (A)	1	1	1	0.98	1	0.17	0.17	0.17	0.18	0.98
Cooperating Servers (B)	1	1	1	1	1	0.17	0.17	0.17	0.18	1
Parallel Queues (C)	1	0.99	0.9	0.76	-	0.17	0.16	0.15	0.14	0.13

In this case, having the smaller stores included in the bigger stores' peer groups make them mostly inefficient.

Table 6.22: Peer Group

A	Peer Group	B	Peer Group	C	Peer Group
1	A_1	1	B_1	1	C_1
2	A_2	2	B_2	2	A_2, A_5, B_3, C_1
3	A_3	3	B_3	3	A_5, B_1, B_3, B_5
4	A_3, B_3, B_4	4	B_4	4	A_5, B_3, B_4, B_5
5	A_5	5	B_5	5	-
6	A_6	6	B_6	6	C_6
7	A_8, B_6	7	B_6, B_9	7	A_9, A_{10} B_6, B_{10}
8	A_8	8	A_6, B_9	8	A_{10}, B_6, B_{10}
9	A_9	9	B_9	9	A_{10}, B_6, B_{10}
10	A_{10}	10	B_{10}	10	A_{10}, B_6, B_{10}

As we can see in the Table 6.22, the stores' Peer Groups do not include stores from different populations. Essentially, the first half remain as before, completely unaffected, and the second half's inefficient ones mainly include A_{10}, B_6 and B_{10}. Therefore, they can be considered the representative Units.

Bibliography

[1] Rajiv D Banker, Abraham Charnes, and William Wager Cooper. "Some models for estimating technical and scale inefficiencies in data envelopment analysis". In: *Management science* 30.9 (1984), pp. 1078–1092.

[2] Rajiv D Banker and Richard C Morey. "The Use of Categorical Variables in Data Envelopment Analysis". In: *Management science* 32.12 (1986), pp. 1613–1627.

[3] Abraham Charnes, William W Cooper, and Edwardo Rhodes. "Measuring the efficiency of decision making units". In: *European journal of operational research* 2.6 (1978), pp. 429–444.

[4] William W Cooper, Lawrence M Seiford, and Kaoru Tone. *Data Envelopment Analysis: a Comprehensive Text with Models, Applications, References and DEA-solver software*. Vol. 2. Springer, 2007.

Alphabetical Index