Chapter 5

Modeling Undesirable Measures

5.1 Introduction

Both desirable (good) and undesirable (bad) outputs and inputs may be present. For example, the number of defective products is an undesirable output. One wants to reduce the number of defects to improve the performance. If inefficiency exists in production processes where final products are manufactured with a production of wastes and pollutants, the outputs of wastes and pollutants are undesirable and should be reduced to improve the performance.

Note that in the conventional DEA models, e.g., the VRS envelopment models, it is assumed that outputs should be increased and the inputs should be decreased to improve the performance or to reach the best-practice frontier. If one treats the undesirable outputs as inputs so that the bad outputs can be reduced, the resulting DEA model does not reflect the true production process.

Situations when some inputs need to be increased to improve the performance are also likely to occur. For example, in order to improve the performance of a waste treatment process, the amount of waste (undesirable input) to be treated should be increased rather than decreased as assumed in the conventional DEA models.

Seiford and Zhu (2002) develop an approach to treat undesirable input/outputs in the VRS envelopment models. The key to their approach is the use of DEA classification invariance under which classifications of efficiencies and inefficiencies are invariant to the data transformation.

5.2 Efficiency Invariance

Suppose that the inputs and outputs are transformed to $\bar{x}_{ij} = x_{ij} + u_i$ and $\bar{y}_{rj} = y_{rj} + v_r$, where u_i and v_r are nonnegative. Then the input-oriented and the output-oriented VRS envelopment models become

$$\begin{aligned}
&\min \theta - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij} + s_i^- = \theta \bar{x}_{io} \qquad i = 1,2,...,m; \\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj} - s_r^+ = \bar{y}_{ro} \qquad r = 1,2,...,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0 \qquad\qquad j = 1,2,...,n.
\end{aligned} \tag{5.1}$$

$$\begin{aligned}
&\max \phi - \varepsilon(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+) \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij} + s_i^- = \bar{x}_{io} \qquad i = 1,2,...,m; \\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj} - s_r^+ = \phi \bar{y}_{ro} \qquad r = 1,2,...,s; \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \geq 0 \qquad\qquad j = 1,2,...,n.
\end{aligned} \tag{5.2}$$

Ali and Seiford (1990) show that DMU_o is efficient under (1.5) or (1.6) if and only if DMU_o is efficient under (5.1) or (5.2). This conclusion is due to the presence of the convexity constraint $\sum_{j=1}^{n} \lambda_j = 1$. This property also enables us to treat possible negative inputs and outputs before applying the VRS model (see Appendix of this chapter.)

In general, there are three cases of invariance under data transformation in DEA. The first case is restricted to the "classification invariance" where the classifications of efficiencies and inefficiencies are invariant to the data transformation. The second case is the "ordering invariance" of the inefficient DMUs. The last case is the "solution invariance" in which the new DEA model (after data translation) must be equivalent to the old one, i.e., both mathematical programming problems must have exactly the same solution. The method of Seiford and Zhu (2002) is concerned only with the first level of invariance – classification invariance. See Pastor (1996) and Lovell and Pastor (1995) for discussions in invariance property in DEA.

5.3 Undesirable Outputs

Let y_{rj}^{g} and y_{rj}^{b} denote the desirable (good) and undesirable (bad) outputs, respectively. Obviously, we wish to increase y_{rj}^{g} and to decrease y_{rj}^{b} to improve the performance. However, in the output-oriented VRS envelopment model, both y_{rj}^{g} and y_{rj}^{b} are supposed to increase to improve the performance. In order to increase the desirable outputs and to decrease the undesirable outputs, we proceed as follows.

First, we multiply each undesirable output by "-1" and then find a proper value v_r to let all negative undesirable outputs be positive. That is, $\bar{y}_{rj}^{b} = -y_{rj}^{b} + v_r > 0$. This can be achieved by $v_r = \max_j \{ y_{rj}^{b} \} + 1$, for example.

Based upon (5.2), we have

$$
\begin{aligned}
&\max h \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j y_{rj}^{g} \geq h y_{ro}^{g} \\
&\sum_{j=1}^{n} \lambda_j \bar{y}_{rj}^{b} \geq h \bar{y}_{ro}^{b} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{io} \\
&\sum_{j=1}^{n} \lambda_j \quad = 1 \\
&\lambda_j \geq 0, \; j = 1, \ldots, n
\end{aligned}
\tag{5.3}
$$

Note that (5.3) increases desirable outputs and decreases undesirable outputs. The following theorem ensures that the optimized undesirable output of y_{ro}^{b} ($= v_r - h^* \bar{y}_{ro}^{b}$) cannot be negative.

Theorem 5.1 Given a translation vector v, suppose h^* is the optimal value to (5.3), we have $h^* \bar{y}_{ro}^{b} \leq v_r$.

[Proof] Note that all outputs now are non-negative. Let λ_j^* be an optimal solution associated with h^*. Since $\sum_{j=1}^{n} \lambda_j^* = 1$, $h^* \bar{y}_{ro}^{b} \leq \bar{y}_r^*$, where $\bar{y}_r^*$ is composed from (translated) maximum values among all bad outputs. Note that $\bar{y}_r^* = -y_r^* + v_r$, where y_r^* is composed from (original) minimum values among all bad outputs. Thus, $h^* \bar{y}_o^{b} \leq v_r$. ■

We may treat the undesirable outputs as inputs. However, this does not reflect the true production process. We may also apply a monotone decreasing transformation (e.g., $1/y_{rj}^{b}$) to the undesirable outputs and then to use the adapted variables as outputs. The current method, in fact, applies a linear monotone decreasing transformation. Since the use of linear transformation preserves the convexity, it is a good choice for a DEA model.

Figure 5.1 illustrates the method. The five DMUs A, B, C, D and E use an equal input to produce one desirable output (g) and one undesirable output (b). GCDEF is the (output) frontier. If we treat the undesirable output as an input, then ABCD becomes the VRS frontier. Model (5.2) rotates the output frontier at EF and obtains the symmetrical frontier. In this case, DMUs A′, B′ and C′, which are the adapted points of A, B and C, respectively, are efficient.

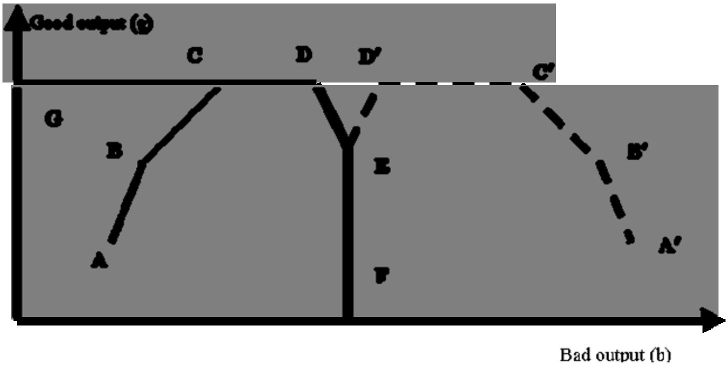

Figure 5.1. Treatment of Bad Output

The efficient target for DMU_o is

$$\begin{cases} \hat{x}_{io} = x_{io} - s_i^{-*} \\ \hat{y}_{ro}^{g} = h^* y_{ro}^{g} + s_r^{+*} \\ \hat{y}_{ro}^{b} = v_r - (h^* \bar{y}_{ro}^{b} + s_r^{+*}) \end{cases}$$

Table 5.1. Vendors

Vendors	Price ($/unit)	%Rejects	% Late deliveries
1	0.1958	1.2	5
2	0.1881	0.8	7
3	0.2204	0	0
4	0.2081	2.1	0
5	0.2118	2.3	3
6	0.2096	1.2	4

Source: Weber and Desai. (1996).

We conclude this section by applying the method to the six vendors studied in Weber and Desai (1996). Table 5.1 presents the data. The input is price per unit, and the outputs are percentage of late deliveries and percentage of rejected units. (See Weber and Desai (1996) for detailed discussion on the input and the two outputs.)

Obviously, the two outputs are bad outputs. We use an translation vector of (3.3%, 8%). (Or one could use (100%, 100%) as in Chapter 7.) Figure 5.2 shows the translated data and the spreadsheet model. This is actually a spreadsheet model for the output-oriented VRS envelopment model. Figure 5.3 shows the Solver parameters. Column G in Figure 5.2 reports the efficiency scores.

	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		2.1	3	0	1.08921
3	Vendor 2	0.1881		2.5	1	0.327059	1
4	Vendor 3	0.2204		3.3	8	0.653754	1
5	Vendor 4	0.2081		1.2	8	0.019187	1
6	Vendor 5	0.2118		1	5	0	1.6
7	Vendor 6	0.2096		2.1	4	0	1.427647
8							
9		Reference		DMU unde	**6**	Efficiency	
10	**Constrain**	set		Evaluation		1.427647	
11	price	0.2096	≤	0.2096			
12	Rejects	2.998059	≥	2.998059	Run		
13	Late delive	5.710589	≥	5.710589			
14	Σλ	1	=	1			

Figure 5.2. Bad Outputs Spreadsheet Model

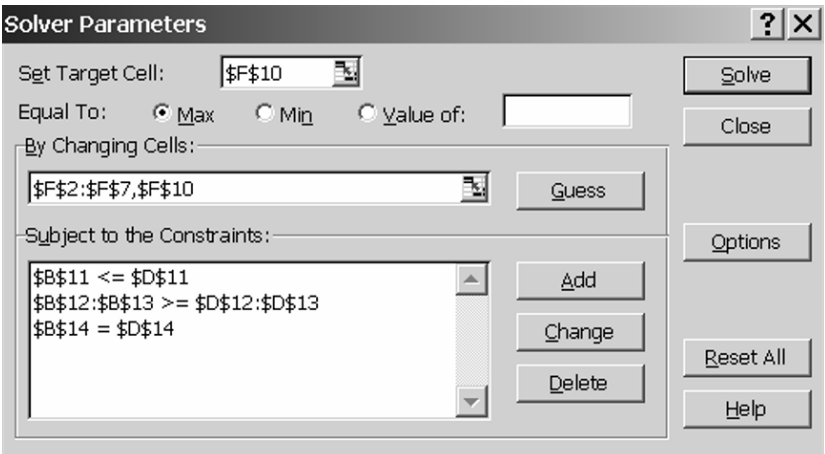

Figure 5.3. Solver Parameters for Bad Outputs Spreadsheet Model

If we do not translate the bad outputs and calculate the regular output-oriented VRS envelopment model, vendor 5 is classified as efficient, and vendor 3 is classified as inefficient. (see Figure 5.4 where 0.0001 is used to replace 0.) The same Solver parameters shown in Figure 5.3 are used.

	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		1.2	5	0	1.075329
3	Vendor 2	0.1881		0.8	7	0.518519	1
4	Vendor 3	0.2204		0.0001	0.0001	0	23000
5	Vendor 4	0.2081		2.1	0	0	1
6	Vendor 5	0.2118		2.3	3	0.481481	1
7	Vendor 6	0.2096		1.2	4	0	1.268519
8							
9		Reference		DMU unde	**6**	Efficiency	
10	**Constrain**	set		Evaluation		1.268519	
11	price	0.199511	≤	0.2096			
12	Rejects	1.522222	≥	1.522222	Run		
13	Late delive	5.074074	≥	5.074074			
14	Σλ	1	=	1			

Figure 5.4. Efficiency Scores When Bad Outputs Are Not Translated

If we treat the two bad outputs as inputs and use the input-oriented VRS envelopment model, we obtain the efficiency scores shown in Figure 5.5 (Figure 5.6 shows the Solver parameters). In this case, we do not have outputs.

	A	B	C	D	E	F	G
1		Price		%Rejects	% Late deliveries	λ	Efficiency
2	Vendor 1	0.1958		1.2	5	0	0.990548
3	Vendor 2	0.1881		0.8	7	0.541904	1
4	Vendor 3	0.2204		0	0	0.122632	1
5	Vendor 4	0.2081		2.1	0	0.335464	1
6	Vendor 5	0.2118		2.3	3	0	0.944315
7	Vendor 6	0.2096		1.2	4	0	0.948332
8							
9		Reference		DMU unde	**6**	Efficiency	
10	**Constrain**	set		Evaluation		0.948332	
11	price	0.19877	≤	0.19877			
12	Rejects	1.137998	≤	1.137998	Run		
13	Late delive	3.793326	≤	3.793326			
14	Σλ	1	=	1			

Figure 5.5. Efficiency Scores When Bad Outputs Are Treated As Inputs

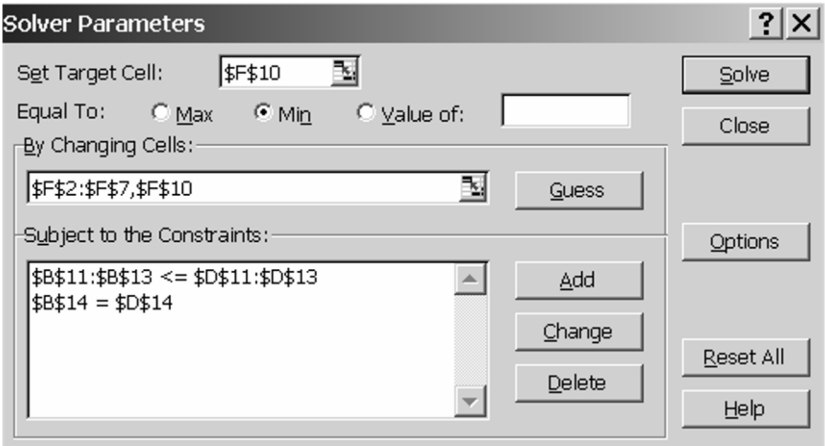

Figure 5.6. Solver Parameters When Bad Outputs Are Treated As Inputs

5.4 Undesirable Inputs

The above discussion can also be applied to situations when some inputs need to be increased rather than decreased to improve the performance. In this case, we denote x_{ij}^I and x_{ij}^D the inputs that need to be increased and decreased, respectively.

We next multiply x_{ij}^I by "-1" and then find a proper u_i to let $\bar{x}_{ij}^I = -x_{ij}^I + u_i > 0$. Based upon model (5.1), we have

$$
\begin{aligned}
&\min \tau \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij}^D \le \tau x_{io}^D \\
&\sum_{j=1}^{n} \lambda_j \bar{x}_{ij}^I \le \tau \bar{x}_{oi}^I \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \quad j=1,\ldots,n.
\end{aligned}
\tag{5.4}
$$

where x_{ij}^I is increased and x_{ij}^D is decreased for a DMU to improve the performance. The efficient target for DMU_o is

$$
\begin{cases}
\hat{x}_{io}^D = \tau^* x_{io}^I - s_i^{-*} \\
\hat{x}_{io}^I = u_i - (\tau^* x_{io}^I - s_i^{-*}) \\
\hat{y}_{ro} = y_{ro} + s_r^{+*}
\end{cases}
$$

5.5 Solving DEA Using DEAFrontier Software

To run the models for treating undesirable measures, select the Undesirable-Measure Model menu item. Figure 5.7 shows the form for specifying the models. The results are reported in "Efficiency", "Slack", and "Target" sheets.

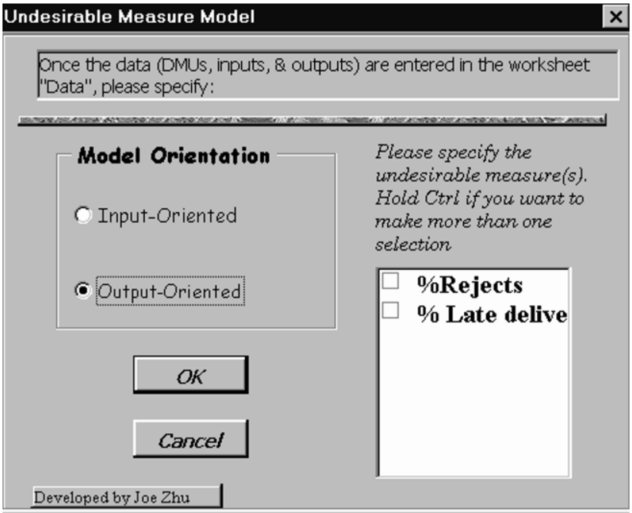

Figure 5.7. Undesirable Measure Models

APPENDIX: NEGATIVE DATA

So far, we have assumed that all inputs and outputs are either positive or zero. However, we have cases where some inputs and (or) outputs are negative. For example, when a company experiences a loss, its profit is negative. Similarly, returns on some stocks can be negative. This can be easily solved by way of the *translation invariance property* of the VRS models (Ali and Seiford, 1990). Specifically, the VRS frontier remains the same if x_{ij} and y_{rj} is replaced to $\bar{x}_{ij}$ and $\bar{y}_{rj}$, respectively.

Consider the example given in the following Table where we have 10 DMUs[1]. We have two inputs x_1 = Standard Deviation and x_2 = PropNeg (proportion of negative monthly returns during the year), and three outputs y_1 = Return (average monthly return), y_2 = Skewness and y_3 = Min (minimum return).

Note that some values for return, skewness and Min are negative. In the table the average monthly return, skewness and minimum return are displaced by 3.7%, 2, and 26%, respectively so that all the output values are positive across all the DMUs. The translation values can be chosen randomly as long as the negative values become positive.

Table A.1 Negative Data Example

DMU	Original Data					Transformed Data $\hat{y}_r = y_{rj} + \pi_r$		
	x_1 = Standard Deviation	x_2 = Proportion Negative	y_1 = Ave. Monthly Return	y_2 = Skewness	y_3 = Minimum Return	$\hat{y}_1$; $\pi_1 = 3.7\%$	$\hat{y}_2$; $\pi_2 = 2$	$\hat{y}_3$; $\pi_3 = 26\%$
1	6.80%	58.30%	0.10%	1.13	-8.10%	3.80%	3.13	17.90%
2	4.00%	41.70%	0.70%	0.61	-7.90%	4.40%	2.61	18.10%
3	3.40%	37.50%	0.90%	0.58	-4.00%	4.60%	2.58	22.00%
4	5.00%	50.00%	0.60%	1.7	-5.60%	4.30%	3.7	20.40%
5	4.70%	37.50%	1.10%	0.28	-8.20%	4.80%	2.28	17.80%
6	3.80%	50.00%	-0.10%	0.08	-6.30%	3.60%	2.08	19.70%
7	11.20%	45.80%	3.20%	0.39	-17.10%	6.90%	2.39	8.90%
8	12.80%	58.30%	-1.00%	0.46	-25.70%	2.70%	2.46	0.30%
9	8.40%	52.20%	-1.20%	-0.26	-17.10%	2.50%	1.74	8.90%
9	5.00%	54.50%	0.40%	1.1	-6.70%	4.10%	3.1	19.30%
10	8.60%	25.00%	-3.60%	-1.98	-16.50%	0.10%	0.02	9.50%

Since negative data are present only in the outputs, we thus use the input-oriented VRS model. When DMU1 is under evaluation, we have $\theta_0^* = 0.75$,

[1] These DMUs are called commodity trading advisors (CTAs) in Wilkens and Zhu (2001).

indicating this DMU is inefficient, and $\lambda_3^* = 0.51$ and $\lambda_4^* = 0.49$, indicating DMU3 and DMU4 are the benchmarks.

REFERENCES

1. Ali, A.I. and L.M. Seiford (1990), Translation invariance in data envelopment analysis, *Operations Research Letters*, 9, 403-405.
2. Lovell, C.A.K. and J.T. Pastor (1995), Units invariant and translation invariant DEA models, *Operations Research Letters*, 18, 147-151.
3. Pastor, J.T. (1996), Translation invariance in data envelopment analysis: a generalization, *Annals of Operations Research*, 66, 93-102.
4. Seiford, L.M. and J. Zhu (2002), Modeling undesirable factors in efficiency evaluation, *European Journal of Operational Research*, Vol. 142, Issue 1, 16-20.
5. Weber, C.A. and A. Desai (1996), Determination of paths to vendor market efficiency using parallel coordinates representation: a negotiation tool for buyers, *European Journal of Operational Research*, 90, 142-155.
6. Wilkens, K. and J. Zhu (2001), Portfolio evaluation and benchmark selection: A mathematical programming approach, *Journal of Alternative Investments*, Vol. 4, No. 1, 9-19