Chapter 2
Data Envelopment Analysis

2.1 Technology

Analysis of performance has economic production theory as its foundation. Firms employ inputs to produce output with an incentive to either maximize profits or minimize costs. Technically inefficient firms could either increase outputs (and therefore revenue) holding inputs constant or could decrease inputs (and hence costs) holding outputs constant. As a result, technically inefficient firms are neither profit maximizing nor cost minimizing. The seminal paper on technical efficiency measurement was Farrell (1957) which provided a decomposition of inefficiency into technical and allocative parts. From an input-oriented perspective, firms that are not operating on the isoquant associated with observed production are technically inefficient. Farrell provided a comprehensive measure of technical efficiency as the equiproportional reduction of all inputs holding output at current levels. Proportional reduction in observed inputs holds the input mix constant. Cost minimization, however, requires not only production on the isoquant but also the appropriate mix of inputs that depends on the associated input prices. Hence, if technically efficient firms are not using the allocatively efficient input mix they could still lower costs by adjusting input levels accordingly.

Farrell provided the formulation to handle a single output in the case of constant returns to scale. The paper also discussed decreasing returns to scale and the extension to multiple outputs. Farrell and Fieldhouse (1962) extended the approach as a linear program allowing increasing returns to scale. Afriat (1972) provided the formulation for technical efficiency measurement that was consistent with data envelopment analysis (DEA). The theoretical foundations of efficiency measurement are provided in Färe, Grosskopf, and Lovell (1994).

DEA is the term coined in the Operations Research literature by Charnes, Cooper, and Rhodes (1978) (CCR) to measure the technical efficiency of a given

V. Blackburn et al., *Nonparametric Estimation of Educational Production and Costs using Data Envelopment Analysis*, DOI 10.1007/978-1-4899-7469-3_2,

observed decision making unit (DMU) assuming constant returns to scale. Their linear programming formulation allowed multiple inputs and multiple outputs. Banker, Charnes, and Cooper (1984) (BCC) extended the CCR model to allow variable returns to scale and showed that solutions to both CCR and BCC allowed a decomposition of CCR efficiency into technical and scale components.

In this section, we introduce the representation of the technology that serves as the basis for efficiency measurement. We assume that each decision making unit (DMU) uses a vector of m discretionary inputs $X = (x_1, \ldots, x_M)$ to produce a vector of s outputs $Y = (y_1, \ldots, y_S)$. Inputs and outputs for DMU_j ($j = 1, \ldots, n$) are given by $X_j = (x_{1j}, \ldots, x_{Mj})$ and $Y_j = (y_{1j}, \ldots, y_{Sj})$. Assuming variable returns to scale, the empirical production possibility set is given by:

$$\begin{aligned} \tau_V = \Big\{(Y,X) : &\sum_{i=1}^{n} \lambda_i y_{si} \geq y_s, \;\; s = 1, \ldots, S; \\ &\sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \;\; m = 1, \ldots, M; \\ &\sum_{i=1}^{n} \lambda_i = 1; \\ &\lambda_i \geq 0, \;\; i = 1, \ldots, N\Big\}. \end{aligned} \tag{2.1}$$

It is assumed that each of the n observed production points is feasible (i.e., standard measurement error and statistical noise does not exist). Further, we assume that any convex combination of the observed points is also feasible and outputs and inputs are freely disposable.

The technology is illustrated in Fig. 2.1, where we assume that four DMUs $A-D$ employ one input x_1 to produce one output y_1. Based on Eq. (2.1), a piecewise linear approximation of the technology emerges. Line segment AB, obtained by convex combinations of A and B, corresponds to increasing returns to scale; likewise, BC (CD) allows constant (decreasing) returns to scale. The variable returns to scale technology results by allowing only convex combinations. The shaded area in Fig. 2.1 reveals the feasible region as defined by Eq. (2.1).

An alternative way to represent the technology is with input requirement sets and isoquants.[1] Following Lovell (1993) we define the input set as

$$L_V(Y) = \{X : (Y,X) \;\in \tau_V\}. \tag{2.2}$$

[1] As is well known in the DEA literature, an inefficient DMU can be on the isoquant. Given that the Farrell measure projects a production possibility to the isoquant, it is possible that the resulting benchmark is not technically efficient. In this case, alternative projections could be used. See Färe and Lovell (1978), Ruggiero and Bretschneider (1998) or Ruggiero (2000) for examples.

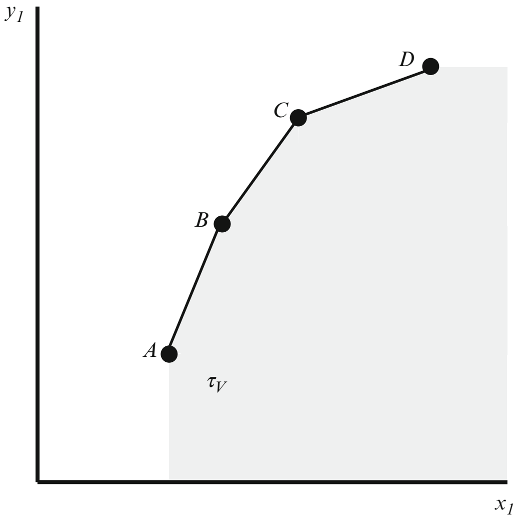


Fig. 2.1 Empirical Production Possibility Set τ_V

For each output vector Y we can define the isoquant for input set $L_V(Y)$ as

$$Isoq\ \ L_V(Y) = \{X : X \in L_V(Y),\ \lambda X \notin L_V(Y),\ \lambda \in [0, 1)\}. \tag{2.3}$$

The isoquant of the input set shows the boundary such that any equiproportional reduction in all inputs cannot produce the given output vector. As such, the isoquant represents the boundary between feasible and infeasible production. The input set and the associated isoquant are useful for illustrating the technology in input space. Following Färe et al. (1994), we specify the input set $L_V(Y)$ using a piecewise linear representation:

$$\begin{aligned} L_V(Y) = \{X : & \sum_{i=1}^{n} \lambda_i y_{si} \geq y_s,\ \ s = 1, \ldots, S; \\ & \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m,\ \ m = 1, \ldots, M; \\ & \sum_{i=1}^{n} \lambda_i = 1; \\ & \lambda_i \geq 0,\ \ i = 1, \ldots, N\}. \end{aligned} \tag{2.4}$$

We illustrate the input set and the associated isoquant in Fig. 2.2, where we assume that four DMUs $A-D$ produce the same level of output Y using two inputs x_1 and x_2. Using Eq. (2.4), we obtain a piecewise linear $Isoq\ L_V(Y)$ defined with line segments AB, BC and CD. The resulting input set $L_V(Y)$ is shown with the shaded area.

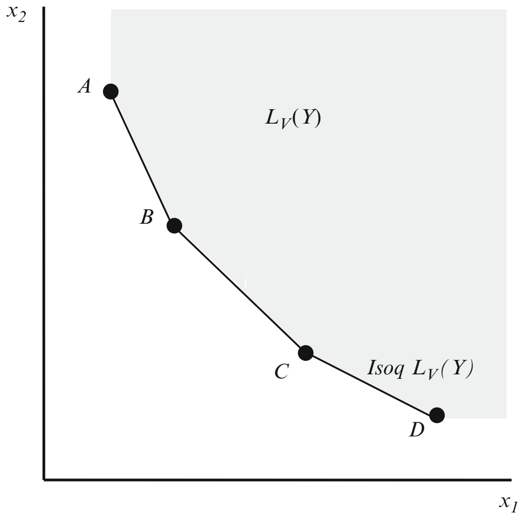


Fig. 2.2 Input Requirement Set $L_V(Y)$and Isoquant *Isoq* $L_V(Y)$

We can also define the technology using the output set

$$P_V(X) = \{Y : (Y, X) \in \tau_V\} \tag{2.5}$$

and its associated isoquant

$$Isoq\ \ P_V(X) = \left\{Y : Y \in P_V(X), \lambda^{-1} Y \notin P_V(X), \lambda \in \left[0, 1\right)\right\}. \tag{2.6}$$

The isoquant of the output set shows the boundary of the output set such that equiproportional expansion of outputs is not feasible without additional resources. The output set and its associated isoquant are useful for illustrating the technology in output space. Following Färe et al. (1994), we specify the input set $P_V(X)$ using a piecewise linear representation:

$$\begin{aligned} P_V(X) = \Big\{Y : &\sum_{i=1}^{n} \lambda_i y_{si} \geq y_s, \quad s = 1, \ldots, S; \\ &\sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \quad m = 1, \ldots, M; \\ &\sum_{i=1}^{n} \lambda_i = 1; \\ &\lambda_i \geq 0, \quad i = 1, \ldots, N\Big\}. \end{aligned} \tag{2.7}$$

The output set and the associated isoquant are illustrated in Fig. 2.3. We assume that four *DMU*s $A - D$ use the same vector of inputs X to produce two outputs y_1 and y_2. Using Eq. (2.7), we obtain a piecewise linear *Isoq* $P_V(X)$ defined with line segments *AB*, *BC* and *CD*. The resulting output set $P_V(X)$ is shown with the shaded area.

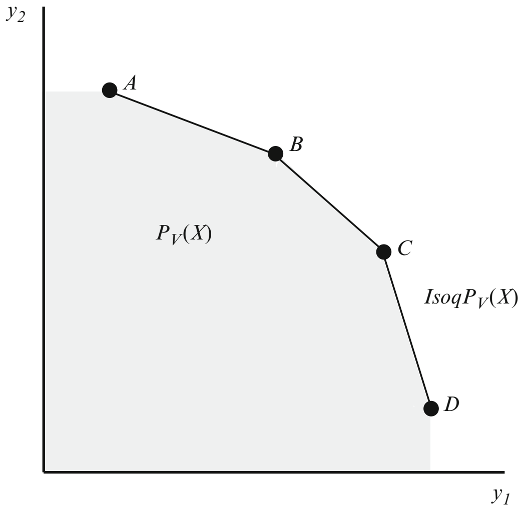


Fig. 2.3 Output Set $P_V(X)$ and Isoquant $Isoq\ P_V(X)$

2.2 Technical Efficiency Measurement

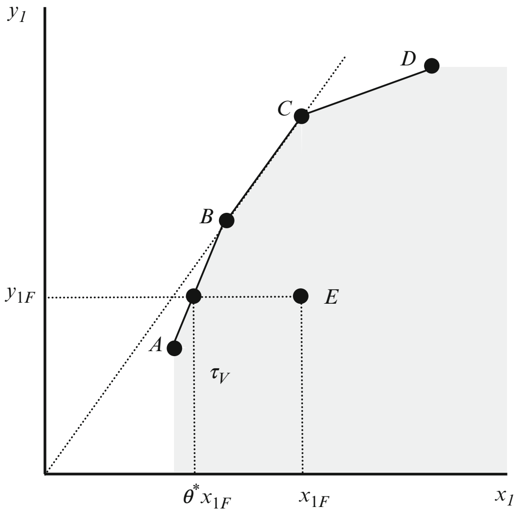
2.2.1 Input-Orientation

Using the technologies defined in Eqs. (2.1) and (2.4) we now consider input-oriented efficiency measures. In this subsection, we consider the Farrell (1957) measure of technical efficiency, defined for DMU_j with the following distance function:

$$D_V^I(Y_j, X_j) = \min\{\theta : (Y_j, \theta X_j) \in \tau_V\}. \tag{2.8}$$

The Farrell measure of technical efficiency projects observed production possibilities to the frontier subject to the constraint that the resulting benchmark projection is feasible, i.e. belongs to τ_V. Hence, efficiency is defined as the maximum equiproportional reduction in observed inputs relative to the variable returns to scale production technology. The technical efficiency measure $D_V^I(Y_j, X_j)$ for each DMU_j ($j = 1, \ldots, n$) is obtained via the solution to the following linear program introduced by Banker et al. (1984) for the multiple input multiple output production correspondence:

Fig. 2.4 Input-Oriented Technical Efficiency



$$
\begin{aligned}
&D_V^I(Y_j, X_j) = \min\theta \\
&\text{subject to} \\
&\qquad \sum_{i=1}^{n} \lambda_i y_{si} \geq y_{sj}, \quad s = 1, \ldots, S; \\
&\qquad \sum_{i=1}^{n} \lambda_i x_{mi} \leq \theta x_{mj}, \quad m = 1, \ldots, M; \\
&\qquad \sum_{i=1}^{n} \lambda_i = 1; \\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, N.
\end{aligned}
\tag{2.9}
$$

The technical efficiency measure is illustrated in Fig. 2.4, where we extend Fig. 2.1 with an additional *DMU F* that is technically inefficient. *DMU F* is observed producing y_{1E} using an input level of x_{1E}. We note that a convex combination of *DMU*s *A* and *B* produces the same output level y_{1E} with less input $\theta^* x_{1E}$. In the solution of linear program 2.9 we find $D_V^I(Y_F, X_F) = \theta^*$. The results indicate that *DMU F* could produce its observed output level using θ^* of its observed input level. For *DMU*s $A-D$, we observe $D_V^I(Y_j, X_j) = 1$; *DMU*s $A-D$ are technically efficient.

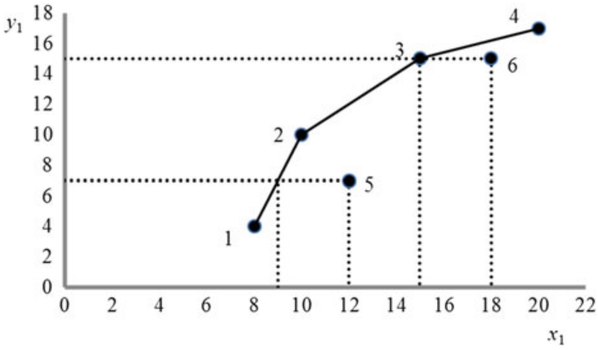


Fig. 2.5 Input-Oriented Technical Efficiency using Example 1 Data

Example 1 Consider the following data where we assume that six *DMU*s employ one input x_1 to produce one output y_1. Data are presented in the following chart.

DMU	x_1	y_1
1	8	4
2	10	10
3	15	15
4	20	17
5	12	7
6	18	15

The data for this example are illustrated in Fig. 2.5. As shown, only *DMU*s 1 – 4 are technically efficient. *DMU* 5 is observed producing 7 units of output using too much x_1. Based on the diagram, we see that *DMU* 5 could have produced the observed output using 9 units of input instead of the observed 12. The benchmark is an equally weighted convex combination of *DMU*s 1 and 2. As a result, the technical efficiency $D_V^I(Y_5, X_5) = \frac{9}{12} = 0.75$; *DMU* 5 should be able to produce the observed 7 units of output using 75 % of the observed 12 units of input. *DMU* 6 is observed using 18 units of the input to produce 15 units of output. Using the input-oriented projection, we observe that technically efficiency *DMU* 3 produces the same output using only 15 units of the input. As shown, *DMU* 3 is the benchmark for *DMU* 6 and $D_V^I(Y_6, X_6) = \frac{15}{18} = 0.8333$. *DMU* 6 is only 83.33 % efficient relative to *DMU* 3.

The SAS code used to measure technical efficiency for example 1 is as follows[2]:

[2] The SAS/OR 12.1 User's Guide Mathematical Programming Examples provides sample DEA code assuming constant returns to scale. The code presented in this book is similar to that code and was provided by SAS.

```
option nonotes;

data example1; input dmu x1 y1;
datalines;
1  8  4
2  10 10
3  15 15
4  20 17
5  12 7
6  18 15
;

proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

   var Weight {DMU} >= 0;
   var theta >= 0;

   min Objective = theta;

   num k;
   num DVI {DMU};
   num benchmark_weight {DMU,DMU};

   con output_con {s in y_Num}:
      sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;

   con input_con {r in x_Num}:
      sum {j in DMU} X[j,r] * Weight[j] <= theta*X[k,r];
```

(continued)

(continued)

```
    con weight_con: sum {j in DMU} Weight[j] = 1;

    do k = DMU;
       solve;
       DVI [k] = theta.sol;
       for {j in DMU} benchmark_weight[k,j] = (if Weight[j].sol > 1e-6 then
           Weight[j].sol else .);
       end;

    create data tech_eff from [dmu] DVI;
    create data benchmark from [dmu dmu_ref] benchmark_weight ;
quit;

data benchmark; set benchmark;
  if benchmark_weight = . then delete;

proc print data = tech_eff;
proc print data = benchmark;
run;
```

The above code produces the following output:

```
                 The SAS System
                  Obs     dmu       DVI
                   1       1      1.00000
                   2       2      1.00000
                   3       3      1.00000
                   4       4      1.00000
                   5       5      0.75000
                   6       6      0.83333

                  The SAS System
                                          benchmark_
            Obs      dmu      dmu_ref       weight
             1        1          1           1.0
             2        2          2           1.0
             3        3          3           1.0
             4        4          4           1.0
             5        5          1           0.5
             6        5          2           0.5
             7        6          3           1.0
```

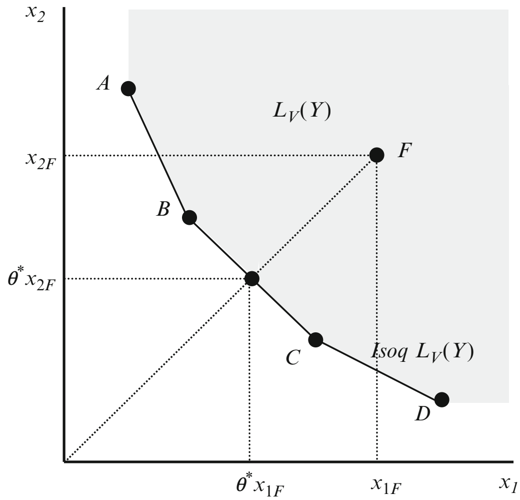


Fig. 2.6 Input-Oriented Efficiency Measurement and *Isoq* $L_V(Y)$

The results are consistent with the discussion above. *DMU*s $1-4$ are technically efficient while *DMU* 5 (6) is only 75 (83.33) percent efficient. In addition, each technically efficient *DMU* has itself as a benchmark. *DMU* 5's benchmark is an equally weighted convex combination of *DMU* 1 and 2 (here labeled as dmu_ref for *DMU* reference). *DMU* 6 is benchmarked against *DMU* 3.

We illustrate efficiency measure using $L_V(Y)$ and its associated isoquant *Isoq* $L_V(Y)$ by extending Fig. 2.2 to add an inefficient *DMU*. In Fig. 2.6, *DMU F* is assumed to produce the same level of output as *DMU*s $A-D$. Here, *DMU F* belongs to the input set but is not on the isoquant. *DMU F* is observed producing the output using inputs x_{1E} and x_{2E}. A convex combination of *DMU*s *B* and *C* produces the same output level with less of both inputs, $\theta^* x_{1E}$ and $\theta^* x_{2E}$. Importantly, the Farrell measure seeks the maximum equiproportional reduction in both inputs. In the solution of the linear program (Eq. (2.9)), $D_V^I(Y_E, X_E) = \theta^*$. The results indicate that *DMU F* could produce its observed output level using θ^* times as much of both inputs. For technically efficient *DMU*s $A-D$, $D_V^I(Y_j, X_j) = 1$.

Example 2 For example 2, we consider a two-input one-output production process where each of five *DMU*s $(1-5)$ produces one unit of output. Data are presented in the following chart. The data are shown in Fig. 2.7.

DMU	x_1	x_2	y_1
1	5	35	1
2	10	20	1
3	20	10	1
4	35	5	1
5	25	25	1

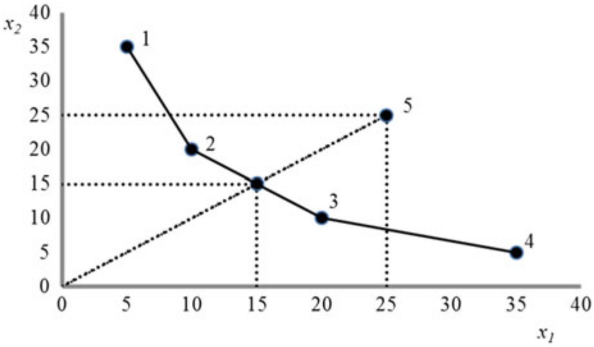


Fig. 2.7 Example 2 Data Input-Oriented Technical Efficiency

As illustrated, the piecewise linear unit isoquant is shown with *DMU*s $1-4$ efficiently producing the observed output. *DMU* 5 is observed producing inefficiently producing the output with 25 units of both inputs. An equally weighted convex combination of technically efficient *DMU*s 2 and 3 uses 15 units of each input. As a result, $D_V^I(Y_5, X_5) = \frac{15}{25} = 0.60$. Hence, DMU 5 is only 60 % efficient.

The SAS code to evaluate the efficiency of all *DMU*s is given below.

```
option nonotes;

data example2; input dmu x1 x2 y1;
datalines;
1 5 35 1
2 10 20 1
3 20 10 1
4 35 5 1
5 25 25 1
;

proc optmodel printlevel = 0;
  set x_num = 1..2;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example2
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

   var Weight {DMU} >= 0;
   var theta >= 0;
```

(continued)

(continued)

```
    min Objective = theta;

    num k;
    num DVI {DMU};
    num benchmark_weight {DMU,DMU};

    con output_con {s in y_Num}:
       sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;

    con input_con {r in x_Num}:
       sum {j in DMU} X[j,r] * Weight[j] <= theta*X[k,r];

    con weight_con: sum {j in DMU} Weight[j] = 1;

    do k = DMU;
       solve;
       DVI [k] = theta.sol;
       for {j in DMU} benchmark_weight[k,j] = (if Weight[j].sol > 1e-6 then
           Weight[j].sol else .);

    end;
    create data tech_eff from [dmu] DVI;
    create data benchmark from [dmu dmu_ref] benchmark_weight ;
quit;

data benchmark; set benchmark;
  if benchmark_weight = . then delete;

proc print data = tech_eff;
proc print data = benchmark;

run;
```

The code after the data are inputted differs from the code for example 1 with the recognition that there are two inputs instead of one (set x_num = 1..2;). This code produces the following SAS output:

The SAS System

Obs	dmu	DVI
1	1	1.0
2	2	1.0
3	3	1.0
4	4	1.0
5	5	0.6

The SAS System

Obs	dmu	dmu_ref	benchmark_ weight
1	1	1	1.0
2	2	2	1.0
3	3	3	1.0
4	4	4	1.0
5	5	2	0.5
6	5	3	0.5

2.2.2 Output-Orientation

We now consider the output-oriented Farrell (1957) measure of technical efficiency using the technologies defined in Eqs. (2.1) and (2.7). We define output-oriented technical efficiency for DMU_j using Shephard's distance function:

$$D_V^O\left(Y_j, X_j\right) = \left(\max\left\{\theta : \left(\theta Y_j, X_j\right) \in \tau_V\right\}\right)^{-1}. \tag{2.10}$$

The Farrell measure of technical efficiency identifies the maximum equiproportional expansion of outputs possible subject to feasibility defined by the technology τ_V. The technical efficiency measure $D_V^O(Y_j, X_j)$ for each DMU_j $(j = 1,\ldots, n)$ is obtained via the solution to the following linear program:

$$\begin{aligned} & D_V^O\left(Y_j, X_j\right)^{-1} = \max\theta \\ & \text{subject to} \\ & \qquad \sum_{i=1}^{n} \lambda_i y_{si} \geq \theta y_{sj}, \quad s = 1, \ldots, S; \\ & \qquad \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_{mj}, \quad m = 1, \ldots, M; \\ & \qquad \sum_{i=1}^{n} \lambda_i = 1; \\ & \qquad \lambda_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{2.11}$$

Fig. 2.8 Output-Oriented Technical Efficiency

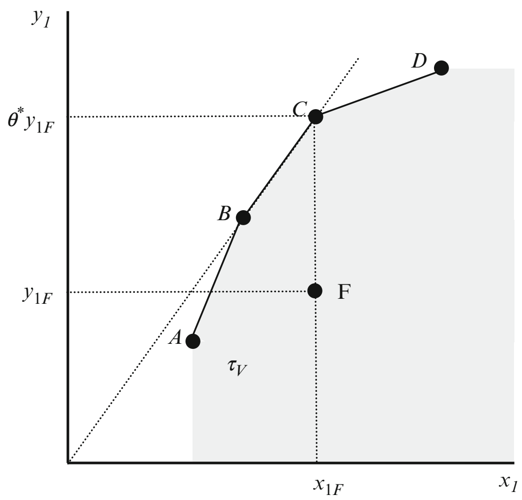
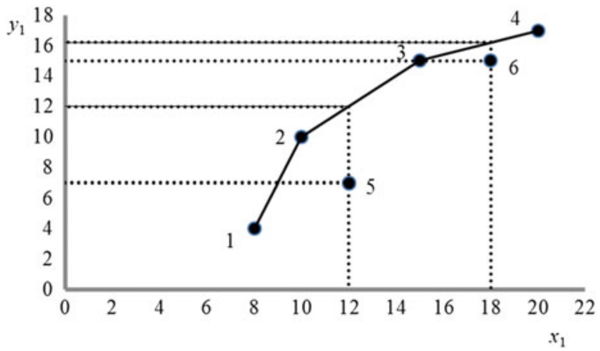


Fig. 2.9 Output-Oriented Technical Efficiency using Example 1 Data



To illustrate the output-oriented measure of technical efficiency, we extend Fig. 2.4. Technically inefficient *DMU F* is observed producing y_{1E} using an input level of x_{1E}. Holding the input level fixed, we seek the maximum expansion in output possible and obtain *DMU C* as the benchmark. DMU C uses the same input level but produces more output $\theta^* y_{1E}$. In the solution of the linear program (Eq. (2.11)) we find $D_V^O(Y_E, X_E) = (\theta^*)^{-1}$. The results indicate that *DMU F* is only producing $(\theta^*)^{-1}$ times as much output as it could be given its observed input usage. Like the input-oriented measures, *DMU*s $A - D$ are technically efficient with $D_V^O(Y_j, X_j)$ equal to unity (Fig. 2.8).

In Fig. 2.5, we illustrated the example 1 data and showed the input-oriented projections used to define the input –oriented technical efficiency measure. Figure 2.9 reproduces the data but shows the output-oriented projections. Two *DMU*s (5 and 6) are technically inefficient using the output-oriented measure of technical efficiency. *DMU* 5 is observed producing only 7 units of output using 12 units of input. The convex combination defined by $\lambda_2 = 0.6$ and $\lambda_3 = 0.4$ uses the same input level but produces 12 units of output. Thus, the convex combination produces 1.71 times as much output. As a result, $D_V^O(Y_5, X_5) = 0.5833$. Hence, *DMU* 5 is only

58.33 % efficient. For *DMU* 6, observed producing 15 units of output while employing 18 units of input, the relevant benchmark is defined by the convex combination $\lambda_3 = 0.4$ and $\lambda_4 = 0.6$. Consequently, we find that $D_V^O(Y_6, X_6) = 0.9259$, implying that *DMU* 6 could increase its output by 8 % given its input usage.

The SAS code used to measure output-oriented efficiency using example 1 data is provided below.

```
option nonotes;

data example1; input dmu x1 y1;
datalines;
1   8   4
2  10  10
3  15  15
4  20  17
5  12   7
6  18  15
;

proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

   var Weight {DMU} >= 0;
   var theta >= 0;

   max Objective = theta;

   num k;
   num DVO {DMU};
   num benchmark_weight {DMU,DMU};

   con output_con {s in y_Num}:
      sum {j in DMU} Y[j,s] * Weight[j] >= theta*Y[k,s] ;

   con input_con {r in x_Num}:
      sum {j in DMU} X[j,r] * Weight[j] <= X[k,r];
```

(continued)

(continued)

```
   con weight_con: sum {j in DMU} Weight[j] = 1;

   do k = DMU;
      solve;
      DVO [k] = 1/theta.sol;
      for {j in DMU} benchmark_weight[k,j] = (if Weight[j].sol > 1e-6 then
Weight[j].sol else .);

   end;
   create data tech_eff from [dmu] DVO;
   create data benchmark from [dmu dmu_ref] benchmark_weight ;
quit;

data benchmark; set benchmark;
  if benchmark_weight = . then delete;

proc print data = tech_eff;
proc print data = benchmark;

run;
```

The resulting SAS output:

```
                 The SAS System
               Obs      dmu        DVO
                1        1       1.00000
                2        2       1.00000
                3        3       1.00000
                4        4       1.00000
                5        5       0.58333
                6        6       0.92593
                  The SAS System
                                          benchmark_
               Obs      dmu      dmu_ref    weight
                1        1          1        1.0
                2        2          2        1.0
                3        3          3        1.0
                4        4          4        1.0
                5        5          2        0.6
                6        5          3        0.4
                7        6          3        0.4
                8        6          4        0.6
```

2.3 Scale Efficiency Measurement

In order to measure scale efficiency and scale economies, we must first define the technology under the assumption of constant returns to scale. Under constant returns to scale, the empirical production possibility set is given by:

$$\begin{aligned}\tau_C = \Big\{(Y, X) : & \sum_{i=1}^{n} \lambda_i y_{si} \geq y_s, \quad s = 1, \ldots, S; \\ & \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \quad m = 1, \ldots, M; \\ & \lambda_i \geq 0, \quad i = 1, \ldots, N\Big\}.\end{aligned} \tag{2.12}$$

Here, we obtain τ_C in Eq. (2.12) from τ_V in Eq. (2.1) by removing the convexity constraint. Likewise, we can define the input requirement set and the output set under constant returns to scale as

$$\begin{aligned}L_C(Y) = \Big\{X : & \sum_{i=1}^{n} \lambda_i y_{si} \geq y_s, \quad s = 1, \ldots, S; \\ & \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \quad m = 1, \ldots, M; \\ & \lambda_i \geq 0, \quad i = 1, \ldots, N\Big\}\end{aligned} \tag{2.13}$$

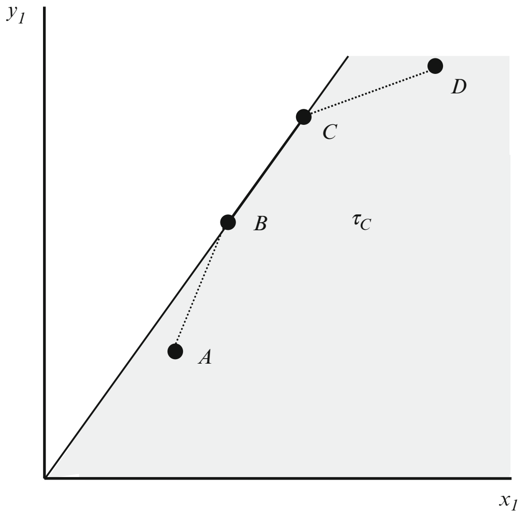
and

$$\begin{aligned}P_C(X) = \Big\{Y : & \sum_{i=1}^{n} \lambda_i y_{si} \geq y_s, \quad s = 1, \ldots, S; \\ & \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \quad m = 1, \ldots, M; \\ & \lambda_i \geq 0, \quad i = 1, \ldots, N\Big\},\end{aligned} \tag{2.14}$$

respectively. Importantly, the frontier defined by Eq. (2.12) allows not only convex combinations but a rescaling of inputs and outputs consistent with constant returns to scale. Necessarily, notions of economies of scale exist only along a production frontier. In order to insure that unit is operating on the variable returns to scale frontier, we can apply either model 2.6 (input-oriented) or 2.11 (output-oriented) to project units to the variable returns to scale frontier and remove any technical inefficiency.

We illustrate the empirical production possibility set based on Eq. (2.12) in Fig. 2.10, which provides the same data points shown in Fig. 2.1. Here, the

Fig. 2.10 Empirical Production Possibility Set τ_C



technology in Eq. (2.12) allows rescaling up or down of *DMU B* (or *DMU C*) along the ray from the origin. If the technology in fact was characterized by constant returns to scale at any level of input usage, then the shaded area would be the feasible region. As shown, *DMU*s *B* and *C* operate on the constant returns to scale portion of the VRS frontier. As a result, both *B* and *C* are maximizing average product. In DEA terms, *B* and *C* are scale efficient, operating at the most productive scale size (see Banker et al., 1984).

*DMU*s *A* and *D*, however, are not operating at most productive scale size. *DMU A* is observed operating under increasing returns to scale could reduce its input level the furthest compared to any point on *AB* while maintaining observed output. Likewise, *DMU D* is operating on the decreasing returns to scale portion of the VRS frontier. Both *DMU*s *A* and *D* are scale inefficient.

2.3.1 Input-Orientation

In order to measure scale efficiency, we need to project a given *DMU* not only to the VRS frontier using Eq. (2.4) but also to the CRS frontier defined in Eq. (2.12). We define the distance function projecting DMU_j to the boundary of the CRS technology with the following distance function:

$$D_C^I(Y_j, X_j) = \min\{\theta : (Y_j, \theta X_j) \in \tau_C\}. \tag{2.15}$$

This associated linear programming model due to Charnes et al. (1978) is given by:

$$
\begin{aligned}
&D_C^I\left(Y_j, X_j\right) = \min \theta_C \\
&\text{subject to} \\
&\qquad \sum_{i=1}^{n} \lambda_i y_{si} \geq y_{sj}, \quad s = 1, \ldots, S; \\
&\qquad \sum_{i=1}^{n} \lambda_i x_{mi} \leq \theta_C x_{mj}, \quad m = 1, \ldots, M; \\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, N.
\end{aligned}
\tag{2.16}
$$

We use θ_C instead of θ to distinguish between the CRS and VRS projections. This model is obtained from Eq. (2.4) by removing the convexity constraint. If the true technology is characterized by variable returns to scale, Eq. (2.16) overestimates true technical inefficiency by projecting to a technically infeasible point if the relevant technically efficient benchmark is characterized by either increasing or decreasing returns to scale. If the technically efficient benchmark is operating under constant returns to scale, the solution of 2.16 is feasible as a solution to 2.4 and technical efficiency is not overestimated.

Banker et al. (1984) introduce the concept of most productive scale size consistent with technically efficient production on the constant returns to scale facet of the production frontier. Technically efficiency production that occurs on increasing (or decreasing) returns to scale facet is not most productive and hence, scale inefficient.[3] In such cases, the solution to (2.16) provides a composed measure of technical and scale inefficiency. A measure of scale efficiency for DMU_j is then obtained as the ratio of two distance functions:

$$
SE^I\left(Y_j, X_j\right) = \frac{D_C^I\left(Y_j, X_j\right)}{D_V^I\left(Y_j, X_j\right)}. \tag{2.17}
$$

A useful interpretation is that the variable returns to scale measure (the denominator) effectively removes technical inefficiency by projecting the unit to the variable returns to scale frontier.

In Fig. 2.11, we show the projection of inefficient *DMU* F to the technology τ_C defined by constant returns to scale. As discussed about Fig. 2.4, *DMU* F is technically inefficient because it could have reduced its input to $\theta^* x_{1E}$ via the solution to Eq. (2.4). The solution of Eq. (2.16) shows the minimum input level $\theta_C^* x_{1E}$ necessary to produce y_{1E} if the technology was characterized by constant returns to scale. However, under the assumption of variable returns to scale, the production possibility $(y_{1E}, \theta^* x_{1E})$ is technically efficient and operating on the

[3] Panzar and Willig (1977) provide a useful discussion of returns to scale in multiple output technologies.

Fig. 2.11 Input-Oriented Projection $D_C^I(y_{1F}, x_{1F})$

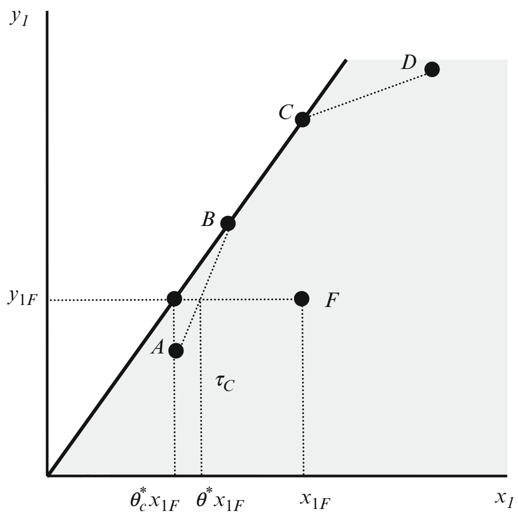
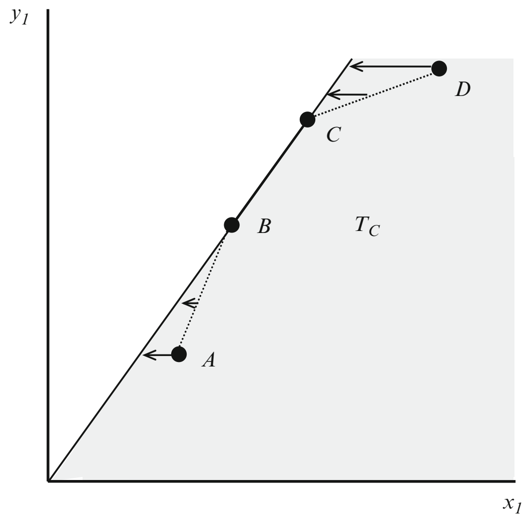


Fig. 2.12 Input Orientation and Scale Efficiency



increasing returns to scale portion of the VRS frontier. The difference in inputs $(\theta^* - \theta_C^*)x_{1E}$ is the amount of extra inputs necessary to produce y_{1E} given that returns are not constant, i.e. $(y_{1E}, \theta^* x_{1E})$ is not most productive scale size. The resulting scale efficiency of DMU F is $SE^I(Y_E, X_E) = \theta_C^*/\theta^* < 1$.

In Fig. 2.12 we show projections from the VRS to the CRS frontiers for four scale inefficient production possibilities. Along the increasing returns to scale portion of the frontier AB, we observe the distance between the VRS and CRS frontiers gets smaller as inputs increase. Along the decreasing returns portion CD the distance gets smaller as we inputs decrease. Hence, our scale efficiency measure increases as we get closer to most productive scale size. Along the constant returns to scale portion *BC* the distance function under CRS and VRS is the same, leading to a scale efficiency measure of unity.

Using data from example 1, we measure the distance functions under both constant and variable returns to scale and calculate the scale efficiency of each unit. The SAS code follows.

```
option nonotes;

data example1; input dmu x1 y1;
datalines;
1  8  4
2  10 10
3  15 15
4  20 17
5  12 7
6  18 15
;

* Projection to the VRS frontier;

proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

   var Weight {DMU} >= 0;
   var theta >= 0;

   min Objective = theta;

   num k;
   num DVI {DMU};
   num benchmark_weight_v {DMU,DMU};

   con output_con {s in y_Num}:
      sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;

   con input_con {r in x_Num}:
      sum {j in DMU} X[j,r] * Weight[j] <= theta*X[k,r];

   con weight_con: sum {j in DMU} Weight[j] = 1;
```

(continued)

(continued)

```
    do k = DMU;
        solve;
        DVI [k] = theta.sol;
        for {j in DMU} benchmark_weight_v[k,j] = (if Weight[j].sol > 1e-6
then Weight[j].sol else .);

    end;
    create data tech_eff_v from [dmu] DVI;
    create data benchmark_v from [dmu dmu_ref] benchmark_weight_v ;
quit;

data benchmark_v; set benchmark_v;
  if benchmark_weight_v = . then delete;

proc print data = benchmark_v;
title 'Benchmarks on the VRS Frontier';

* Projection to the CRS frontier;

proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

    var Weight {DMU} >= 0;
    var theta >= 0;

    min Objective = theta;

    num k;
    num DCI {DMU};
    num benchmark_weight_c {DMU,DMU};

    con output_con {s in y_Num}:
        sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;
```

(continued)

(continued)

```
    con input_con {r in x_Num}:
        sum {j in DMU} X[j,r] * Weight[j] <= theta*X[k,r];

    do k = DMU;
        solve;
        DCI [k] = theta.sol;
        for {j in DMU} benchmark_weight_c[k,j] = (if Weight[j].sol > 1e-6
then Weight[j].sol else .);

    end;
    create data tech_eff_c from [dmu] DCI;
   create data benchmark_c from [dmu dmu_ref] benchmark_weight_c ;
quit;

data benchmark_c; set benchmark_c;
  if benchmark_weight_c = . then delete;
proc print data = benchmark_c;
title 'Benchmarks on the CRS Frontier';
proc sort; by dmu;
proc means sum noprint; var benchmark_weight_c; by dmu; output out =
crs_sum_of_weights sum = sum_weights;

data crs_sum_of_weights; set crs_sum_of_weights;
 keep dmu sum_weights;


proc sort data = tech_eff_v; by dmu;
proc sort data = tech_eff_c; by dmu;
proc sort data = crs_sum_of_weights; by dmu;

data final; merge tech_eff_v tech_eff_c crs_sum_of_weights; by dmu;
  scale_eff = DCI/DVI;
  If scale_eff > 0.9999999 then rts_class = "CRS"; else if
   sum_weights < 1 then rts_class = "IRS"; else rts_class = "DRS";
proc print data = final;
title 'Final Results';

run;
```

The SAS output from the above code:

Benchmarks on the VRS Frontier

Obs	dmu	dmu_ref	benchmark_ weight_v
1	1	1	1.0
2	2	2	1.0
3	3	3	1.0
4	4	4	1.0
5	5	1	0.5
6	5	2	0.5
7	6	3	1.0

Benchmarks on the CRS Frontier

Obs	dmu	dmu_ref	benchmark_ weight_c
1	1	2	0.4
2	2	2	1.0
3	3	2	1.5
4	4	2	1.7
5	5	2	0.7
6	6	2	1.5

Final Results

Obs	dmu	DVI	DCI	sum_ weights	scale_ eff	rts_ class
1	1	1.00000	0.50000	0.4	0.50000	IRS
2	2	1.00000	1.00000	1.0	1.00000	CRS
3	3	1.00000	1.00000	1.5	1.00000	CRS
4	4	1.00000	0.85000	1.7	0.85000	DRS
5	5	0.75000	0.58333	0.7	0.77778	IRS
6	6	0.83333	0.83333	1.5	1.00000	CRS

The results provided by SAS are illustrated in Fig. 2.13. *DMU* 2 was the benchmark for all *DMU*s in the projections to the CRS frontier. Given that *DMU* 2 is technically efficient and operating on the CRS portion of the VRS frontier, its production is scaled down (for the increasing returns to scale *DMU*s) or up (for the decreasing returns to scale *DMU*s). The solution, however, is not unique.

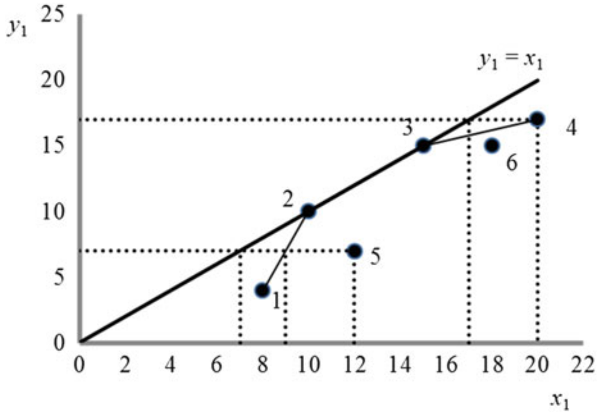


Fig. 2.13 Input-Oriented Scale Efficiency Using Example 1 Data

DMU 3, which is also most productive scale size, could have been rescaled as well. The input levels associated with CRS are determined by the line from the origin through *DMU*s 2 and 3 (the 45° line $y_1 = x_1$.)

Given the distance functions, we calculate the scale efficiency as the ratio of the distance functions. For *DMU* 1, which is technically efficiency, we find $SE^I(Y_1, X_1) = 0.5/1 = 0.5$. If constant returns to scale did exist, *DMU* 1 would be able to produce 4 units of output using only 4 units of input. This ratio of the distance functions is equivalent to the ratio of the input level (4) needed to produce the observed output (4) to the technically efficient input level (8). The results also reveal that *DMU*s 2 and 3 are both technically and scale efficient. *DMU* 6 is technically inefficient but would have been operating at most productive scale size after technically efficiency was eliminated via the input oriented model.

The results also provide information not only about scale efficiency but also about the returns to scale classification. A benchmark[4] is operating under constant returns to scale using the input oriented model if $S^I(Y_j, X_j) = 1$. If the benchmark is scale inefficient, $\sum_{j=1}^{n} \lambda_j^*$ obtained in the solution of (2.16) provides information on the scale class; if $\sum_{j=1}^{n} \lambda_j^* < 1$, the benchmark is operating on the increasing returns to scale portion of the frontier. Here, a most productive scale size production possibility is being scaled downward below the constant returns to scale frontier. If $\sum_{j=1}^{n} \lambda_j^* > 1$, the benchmark is operating under decreasing returns to scale because

[4] We refer only to the benchmark to reinforce the notion that returns to scale is not identified for technically inefficient units.

a most productive scale size production possibility is being rescaled beyond constant returns to scale. The results from example 1 show that the condition on the sum of weights only applies to the scale inefficient units. The solution of 2.16 for *DMU* 3 was obtained by rescaling *DMU* 2 up. But this solution is not unique. *DMU* 3 could have served as its own benchmark while obtaining the same objective value.

2.3.2 Output-Orientation

Alternatively, we can measure scale efficiency using the output oriented model. The measures are similar; we need to project each *DMU* to both the VRS and CRS frontiers. For completeness, we define the distance function projecting DMU_j to the boundary of the CRS technology using an output-orientation:

$$D_C^O\left(Y_j, X_j\right) = \left(\max\left\{\theta : \left(\theta Y_j, X_j\right) \in \tau_C\right\}\right)^{-1}. \tag{2.18}$$

This associated linear programming model to estimate this distance function is:

$$\begin{aligned} & D_C^O\left(Y_j, X_j\right)^{-1} = \max \theta_C \\ & \text{subject to} \\ & \qquad \sum_{i=1}^{n} \lambda_i y_{si} \geq \theta_C y_{sj}, \quad s = 1, \ldots, S; \\ & \qquad \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_{mj}, \quad m = 1, \ldots, M; \\ & \qquad \lambda_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{2.19}$$

This model is the output-oriented equivalent model where the convexity constraint has been removed from Eq. (2.11). Similar to the procedure for the input-oriented model, we can estimate scale efficiency as the ratio of distance functions:

$$SE^O\left(Y_j, X_j\right) = \frac{D_C^O\left(Y_j, X_j\right)}{D_V^O\left(Y_j, X_j\right)}. \tag{2.20}$$

The output-oriented projection of inefficient *DMU* F to the technology τ_C defined by constant returns to scale is illustrated in Fig. 2.14. *DMU* F is technically inefficient because it could have increased its output to $\theta_C^* y_{1E}$ (with $\theta_C^* > 1$) holding input at the observed level x_{1E} in the solution to Eq. (2.19). The benchmark projection *DMU C* (from the solution to Eq. (2.11)) is operating under constant returns to scale and hence, $\lambda_C = 1$ in a solution to 2.19. Since returns to scale are

Fig. 2.14 Output-Oriented Projection $D_C^O(y_{1E}, x_{1F})$

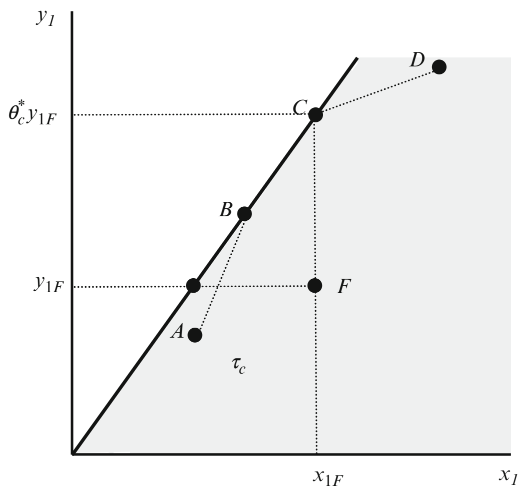
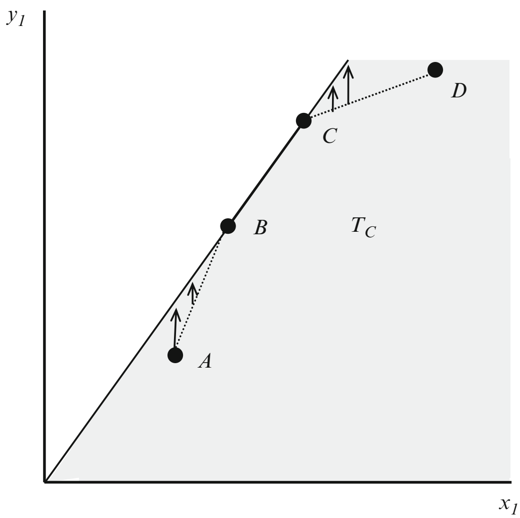


Fig. 2.15 Output Orientation and Scale Efficiency



classified on the frontier, after technical efficiency is eliminated, the resulting scale efficiency of *DMU F* is $SE^O(Y_E, X_E) = \theta_C^* / \theta_C^* = 1$. Also, for DMU EWe also infer from Fig. 2.14 that $SE^O(Y_j, X_j) < 1$ for *DMU*s *A* and *D* and that $SE^O(Y_j, X_j) = 1$ for most productive scale size *DMU*s *B* and *C*.

Like we showed in Fig. 2.12 for the input-oriented model, we show output-oriented projections from the VRS to the CRS frontiers for four scale inefficient production possibilities in Fig. 2.15. Along the increasing returns to scale portion of the frontier *AB*, we observe the distance between the VRS and CRS frontiers gets smaller as inputs increase. Along the decreasing returns portion *CD* the distance gets smaller as we inputs decrease. Hence, the scale efficiency measure using an output-oriented projection behaves the same as it does in the input-oriented case: scale efficiency increases as we get closer to most productive scale size.

We measure the scale efficiency using the output-oriented projections using the data from example 1. The SAS code to estimate the relevant distance functions, the scale efficiency and the returns to scale class follows.

```
option nonotes;
data example1; input dmu x1 y1;
datalines;
1  8  4
2  10 10
3  15 15
4  20 17
5  12 7
6  18 15
;
* Projection to the VRS frontier;
proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;
   var Weight {DMU} >= 0;
   var theta >= 0;
```

(continued)

(continued)

```
    max Objective = theta;

    num k;
    num DVO {DMU};
    num benchmark_weight_v {DMU,DMU};

    con output_con {s in y_Num}:
       sum {j in DMU} Y[j,s] * Weight[j] >= theta*Y[k,s] ;

    con input_con {r in x_Num}:
       sum {j in DMU} X[j,r] * Weight[j] <= X[k,r];

    con weight_con: sum {j in DMU} Weight[j] = 1;

    do k = DMU;
       solve;
       DVO [k] = 1/theta.sol;
       for {j in DMU} benchmark_weight_v[k,j] = (if Weight[j].sol > 1e-6
then Weight[j].sol else .);

    end;
    create data tech_eff_v from [dmu] DVO;
    create data benchmark_v from [dmu dmu_ref] benchmark_weight_v ;
quit;

data benchmark_v; set benchmark_v;
  if benchmark_weight_v = . then delete;

proc print data = benchmark_v;
title 'Benchmarks on the VRS Frontier';

* Projection to the CRS frontier;

proc optmodel printlevel = 0;
  set x_num = 1..1;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example1
    into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

    var Weight {DMU} >= 0;
    var theta >= 0;
```

(continued)

(continued)

```
    max Objective = theta;

    num k;
    num DCO {DMU};
    num benchmark_weight_c {DMU,DMU};

    con output_con {s in y_Num}:
        sum {j in DMU} Y[j,s] * Weight[j] >= theta*Y[k,s] ;

    con input_con {r in x_Num}:
        sum {j in DMU} X[j,r] * Weight[j] <= X[k,r];

    do k = DMU;
        solve;
        DCO [k] = 1/theta.sol;
        for {j in DMU} benchmark_weight_c[k,j] = (if Weight[j].sol > 1e-6
then Weight[j].sol else .);

    end;
    create data tech_eff_c from [dmu] DCO;
   create data benchmark_c from [dmu dmu_ref] benchmark_weight_c ;
quit;

data benchmark_c; set benchmark_c;
  if benchmark_weight_c = . then delete;
proc print data = benchmark_c;
title 'Benchmarks on the CRS Frontier';
proc sort; by dmu;
proc means sum noprint; var benchmark_weight_c; by dmu; output out =
crs_sum_of_weights sum = sum_weights;

data crs_sum_of_weights; set crs_sum_of_weights;
 keep dmu sum_weights;

proc sort data = tech_eff_v; by dmu;
proc sort data = tech_eff_c; by dmu;
proc sort data = crs_sum_of_weights; by dmu;

data final; merge tech_eff_v tech_eff_c crs_sum_of_weights; by dmu;
  scale_eff = DCO/DVO;
  If scale_eff > 0.9999999 then rts_class = "CRS"; else if
    sum_weights < 1 then rts_class = "IRS"; else rts_class = "DRS";
proc print data = final;
title 'Final Results';

run;
```

The SAS output from the above code:

```
            Benchmarks on the VRS Frontier

                                        benchmark_
         Obs      dmu      dmu_ref       weight_v

          1        1          1            1.0
          2        2          2            1.0
          3        3          3            1.0
          4        4          4            1.0
          5        5          2            0.6
          6        5          3            0.4
          7        6          3            0.4
          8        6          4            0.6

            Benchmarks on the CRS Frontier

                                        benchmark_
         Obs      dmu      dmu_ref       weight_c

          1        1          2            0.8
          2        2          2            1.0
          3        3          2            1.5
          4        4          2            2.0
          5        5          2            1.2
          6        6          2            1.8

                      Final Results

                                         sum_     scale_    rts_
   Obs    dmu      DVO        DCO       weights    eff     class

    1      1     1.00000    0.50000      0.8       0.50     IRS
    2      2     1.00000    1.00000      1.0       1.00     CRS
    3      3     1.00000    1.00000      1.5       1.00     CRS
    4      4     1.00000    0.85000      2.0       0.85     DRS
    5      5     0.58333    0.58333      1.2       1.00     CRS
    6      6     0.92593    0.83333      1.8       0.90     DRS
```

The data and projections reported in the SAS output above are illustrated in Fig. 2.16. Like the case for the input-oriented projections, *DMU* 2 serves as the benchmark for all *DMU*s in the projections to the CRS frontier. Based on the projections, we see that only DMU 1 is operating under increasing returns to scale. DMUs 2, 3 and 5 are operating under constant returns to scale. Only DMUs 2 and 3 are technically and scale efficient. DMU 5 is technically efficient; unlike the input-oriented projection, DMU 5 is projected to the CRS portion of the

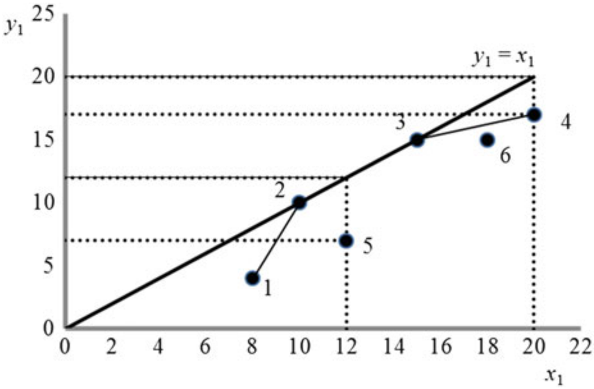


Fig. 2.16 Output-Oriented Scale Efficiency Using Example 1 Data

VRS frontier. As a result, DMU 5 is identified as technically inefficient but scale efficient in the output oriented model. The change in classification happens with DMU 6 as well; under the input-oriented model, it was projected to the constant returns to scale portion of the VRS frontier. In the output-oriented models, however, it is projected to the DRS portion.

Similar to the input projections, we note that the returns to scale classification is obtained from the scale efficiency measure and the sum of the weights in the solution of Eq. (2.19). A benchmark is operating under constant returns to scale using the input oriented model if $S^O(Y_j, X_j) = 1$. If the benchmark is scale inefficient, $\sum_{j=1}^{n} \lambda_j^*$ obtained in the solution of (2.19) provides information on the scale class; if $\sum_{j=1}^{n} \lambda_j^* < 1,$ the benchmark is operating on the increasing returns to scale portion of the frontier. Here, a most productive scale size production possibility is being scaled downward below the constant returns to scale frontier. If $\sum_{j=1}^{n} \lambda_j^* > 1$, the benchmark is operating under decreasing returns to scale because a most productive scale size production possibility is being rescaled beyond constant returns to scale.

2.4 Allocative Efficiency Measurement

To this point, we have presented DEA models employing distance functions to measure technical efficiency and returns to scale using only quantity measures of inputs and outputs. The Farrell measure of technical efficiency holds input and output mixes constant by seeking equiproportional reduction (expansion) of inputs (outputs). But a DMU that is operating technically efficiently might be choosing the wrong mix of inputs or outputs depending on the prevailing prices. Consequently,

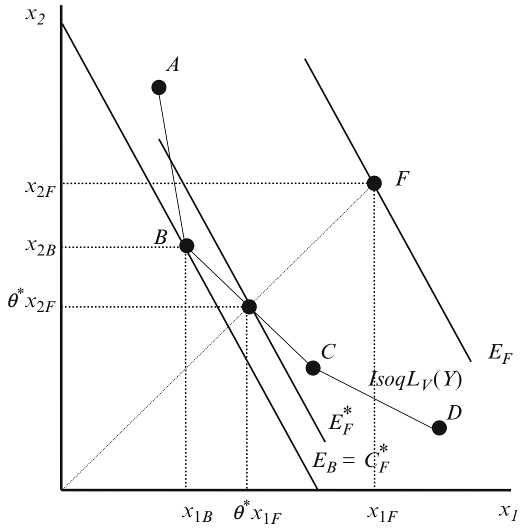


Fig. 2.17 Allocative and Technical Efficiency

the allocatively inefficient *DMU* is spending above minimum costs.[5] Farrell (1957) provided a decomposition of overall efficiency into technical and allocative parts. In this section, we focus on cost efficiency when input prices are available.[6]

Maintaining the notation defined in Sect. 2.1, we now assume that each decision making unit uses a vector of m discretionary inputs $X = (x_1, \ldots, x_M)$ to produce a vector of s outputs $Y = (y_1, \ldots, y_S)$ while facing exogenous input prices $P = (p_1, \ldots, p_M)$. We represent the inputs, outputs and prices for DMU_j $(j = 1, \ldots, n)$ as $X_j = (x_{1j}, \ldots, x_{Mj})$, $Y_j = (y_{1j}, \ldots, y_{Sj})$ and $P_j = (p_{1j}, \ldots, p_{Mj})$. Given the observed input and the associated prices, observed expenditures E_j for DMU_j $(j = 1, \ldots, n)$ can be calculated as $E_j = \sum_{l=1}^{M} p_{lj} x_{lj}$. Feasibility of a production possibility is still subject to the piecewise linear approximation defined in Sect. 2.1.

In Fig. 2.17, we consider five DMUs observed producing the same output levels. Here, DMUs $A-D$ are technically efficient, producing the producing output on the isoquant *Isoq* $L_V(Y)$. Projection possibility F is technically inefficient producing the same output with $D_V^I(Y_F, X_F) = \theta*$. WOLOG, assume that all of the DMUs face the same prices $P_F = (p_{1F}, p_{2F})$. Given the observed prices, we show three isocost lines: E_F, E_F^* and C_F^*. The first isocost line $E_F = p_{1F}x_{1F} + p_{2F}x_{2F}$ shows the observed

[5] Much of the material in this section (and previous sections) is formally derived by Färe et al. (1994).

[6] The extension to revenue efficiency and output orientation is similar. In our empirical applications, we do not have output prices and hence, omit this discussion. See Färe et al. (1994) for a complete treatment.

expenditures of $DMU\ F$. There are two sources of increased spending above the minimum cost level. The first is due to technical inefficiency. The second isocost line $E_F^* = \theta^*(p_{1F}x_{1F} + p_{2F}x_{2F}) = \theta^* E_F$ is the spending that would result if $DMU\ F$ were technically efficient, where θ^* is obtained in the solution of Eq. (2.9). Finally, the isocost line $C_F^* = p_{1F}x_{1B} + p_{2F}x_{2B} = E_B$ represents the minimum cost of producing the observed output given the observed prices. $DMU\ B$ is the only feasible production possibility that can produce the observed output at the cost level C_F^*.

We can define the level of cost efficiency (CE) as the ratio of minimum cost to observed expenditures. For $DMU\ F$, $CE = C_F^*/E_F$ is a measure of cost efficiency. As shown, the cost inefficiency of F arises for two reasons; producing off of the isoquant (technical inefficiency) and using the wrong mix of inputs (allocative inefficiency) given the observed prices. Farrell (1957) provided a decomposition of the overall cost inefficiency into technical and scale components.

To obtain the measure cost efficiency, we first solve the following linear programming model for the minimum cost of producing the observed output for $DMU\ j$ as:

$$
\begin{aligned}
C_j^* &= \min \sum_{m=1}^{M} p_{mj} x_m \\
&\text{subject to} \\
&\quad \sum_{i=1}^{n} \lambda_i y_{si} \geq y_{sj}, \quad s = 1, \ldots, S; \\
&\quad \sum_{i=1}^{n} \lambda_i x_{mi} \leq x_m, \quad m = 1, \ldots, M; \\
&\quad \sum_{i=1}^{n} \lambda_i = 1; \\
&\quad \lambda_i \geq 0, \quad i = 1, \ldots, N.
\end{aligned}
\tag{2.21}
$$

We assume a variable returns to scale technology with the observed convexity constraint and obtain an optimal vector of inputs $(x_1^*, \ldots, x_M^*)$ for each DMU that minimizes the costs of production. Our measure of cost efficiency is then derived as the ratio of minimum costs to observed expenditures:

$$CE\left(Y_j, X_j\right) = \frac{C_j^*}{E_j}. \tag{2.22}$$

Finally, given our cost efficiency measure $CE(Y_j, X_j)$ and our technical efficiency measure $D_V^I(Y_j, X_j)$ we can obtain a measure of allocative efficiency as:

$$AE\left(Y_j, X_j\right) = \frac{CE\left(Y_j, X_j\right)}{D_V^I\left(Y_j, X_j\right)}. \tag{2.23}$$

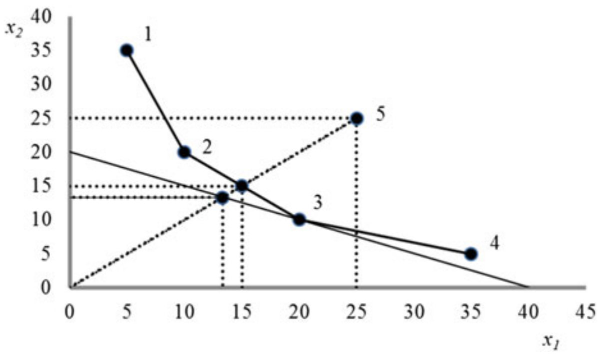


Fig. 2.18 Example 2 Data Technical and Allocative Efficiency

For DMU F in Fig. 2.17, we observe $AE(Y_F, X_F) = C_F^*/E_F^*$.

We extend the data in Example 2 to include input prices and observed expenditures:

DMU	x_1	x_2	p_1	p_2	y_1	E
1	5	35	5	10	1	375
2	10	20	5	10	1	250
3	20	10	5	10	1	200
4	35	5	5	10	1	225
5	25	25	5	10	1	375

These data are presented in Fig. 2.18, where we focus on the technical and allocative efficiency of *DMU* 5, which is observed producing the unit output with 25 units of both inputs. Given the observed prices of $p_1 = 5$ and $p_2 = 5$ *DMU* 5 is observed spending \$375. If *DMU* 5 eliminated its technical inefficiency, it could produce on the isoquant using 15 units of both inputs at a cost of \$225. This holds the input mix $x_2/x_1 = 1$ constant but does not minimize the cost of producing the output. As observed from the table and the SAS output, *DMU* 3 serves as a benchmark for *DMU* 5 (and all of the *DMU*s). *DMU* 3 produces the same output with expenditures of \$200 using an input mix of $x_2/x_1 = 0.5$ (the ratio of the p_1/p_2.) This example also illustrates the problem of using only technical efficiency as a benchmark. *DMU* 1 is technically efficient in production but spends the same amount on inputs as *DMU* 5. For DMU 5, we observe $D_V^I(Y_5, X_5) = \frac{15}{25}$, $CE(Y_5, X_5) = \frac{13.33}{25} = 0.5333$ and $AE(Y_5, X_5) = 0.89$.

The SAS code to estimate cost and allocative efficiency for all *DMU*s is provided below.

```
option nonotes;

data example2; input DMU x1 x2 p1 p2 y1 E;
datalines;
1   5   35   5  10  1   375
2  10   20   5  10  1   250
3  20   10   5  10  1  200
4  35    5   5  10   1  225
5  25   25  5  10   1  375
;

* estimating technical efficiency;

proc optmodel printlevel = 0;
  set x_num = 1..2;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num Y {DMU, y_num};
  read data example2
    into DMU = [dmu]
   {r in x_num} < X[dmu, r] = col("x"||r)>
   {s in y_num} < Y[dmu, s] = col("y"||s)>;

   var Weight {DMU} >= 0;
   var theta >= 0;
  min Objective = theta;
  num k;
  num DVI {DMU};
  num benchmark_weight {DMU,DMU};

  con output_con {s in y_Num}:
  sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;

  con input_con {r in x_Num}:
    sum {j in DMU} X[j,r] * Weight[j] <= theta*X[k,r];

  con weight_con: sum {j in DMU} Weight[j] = 1;
    do k = DMU;
      solve;
      DVI [k] = theta.sol;
     for {j in DMU} benchmark_weight[k,j] = (if Weight[j].sol > 1e-6 then
        Weight[j].sol else .);
    end;
```

(continued)

(continued)

```
    create data tech_eff from [dmu] DVI;
   create data benchmark from [dmu dmu_ref] benchmark_weight ;
quit;

* estimating minimum costs;

proc optmodel printlevel = 0;
  set x_num = 1..2;
  set y_num = 1..1;
  set <num> DMU;
  num X {DMU, x_num};
  num P {DMU, x_num};
  num Y {DMU, y_num};
  read data example2
     into DMU = [dmu]
    {r in x_num} < X[dmu, r] = col("x"||r)>
    {r in x_num} < P[dmu, r] = col("p"||r)>
    {s in y_num} < Y[dmu, s] = col("y"||s)>;

 var Weight {DMU} >= 0;
 var xo{x_num} >= 0;

 num k;
 min Objective = sum{t in x_num} xo[t]*P[k,t];
 num C{DMU};
 num c_benchmark_weight {DMU,DMU};

 con output_con {s in y_Num}:
   sum {j in DMU} Y[j,s] * Weight[j] >= Y[k,s] ;

 con input_con {r in x_Num}:
   sum {j in DMU} X[j,r] * Weight[j] <= xo[r];

 con weight_con: sum {j in DMU} Weight[j] = 1;

 do k = DMU;
    solve;
    C[k] = sum{t in x_num} xo[t]*P[k,t];
    for {j in DMU} c_benchmark_weight[k,j] = (if Weight[j].sol > 1e-6 then
       Weight[j].sol else .);
 end;
```

(continued)

(continued)

```
  create data cost from [dmu] C;
  create data c_benchmark from [dmu dmu_ref] c_benchmark_weight ;
quit;

data c_benchmark; set c_benchmark;
   if c_benchmark_weight = . then delete;

proc print data=c_benchmark;

proc sort data = example2; by dmu;
proc sort data = tech_eff; by dmu;
proc sort data = cost; by dmu;

data final; merge example2 tech_eff cost; by dmu;
  TE = DVI;
  CE = C/E;
  AE = CE/DVI;
proc print data = final; var DMU C TE AE CE;
run;
```

The resulting SAS output:

```
                        The SAS System

                                          c_benchmark_
          Obs     dmu     dmu_ref            weight

           1       1         3                 1
           2       2         3                 1
           3       3         3                 1
           4       4         3                 1
           5       5         3                 1

                        The SAS System

      Obs     DMU     C      TE      AE         CE
       1       1     200    1.0    0.53333    0.53333
       2       2     200    1.0    0.80000    0.80000
       3       3     200    1.0    1.00000    1.00000
       4       4     200    1.0    0.88889    0.88889
       5       5     200    0.6    0.88889    0.53333
```

In this chapter, we defined technologies and presented the standard data envelopment models using piecewise linear frontiers. Linear programming models were developed to estimate technical, scale and allocative efficiency using an input-oriented framework. Output-oriented technical and scale efficiency measures were also presented. In addition, SAS code was provided for all measures and implemented using two illustrative data sets. In the next chapter, we extend these models to allow environmental control variables useful for public sector applications.

http://www.springer.com/978-1-4899-7468-6

Nonparametric Estimation of Educational Production
and Costs using Data Envelopment Analysis
Blackburn, V.; Brennan, S.; Ruggiero, J.
2014, X, 152 p. 34 illus., 10 illus. in color., Hardcover
ISBN: 978-1-4899-7468-6