Chapter 2

Multiplier and Slack-based Models

2.1 Multiplier Model with Weight Restrictions

The dual linear programming problems to the envelopment models are called multiplier models as shown in Table 2.1.

The dual variables v_i and μ_r are called multipliers. A DMU is on the frontier if and only if $\sum_{r=1}^{s}\mu_r y_{ro}$ + μ = 1 (or $\sum_{i=1}^{m} v_i x_{io}$ + v = 1) in optimality. The ε in the envelopment model essentially requires that v_i and μ_r are positive in the multiplier models. The constraint $\sum_{i=1}^{m} v_i x_{io}$ = 1 (or $\sum_{r=1}^{s}\mu_r y_{ro}$ = 1) is known as a normalization constraint. In DEA, the weighted input and output of $\sum_{i=1}^{m} v_i x_{ij}$ and $\sum_{r=1}^{s}\mu_r y_{rj}$ are called virtual input and virtual output, respectively. See Seiford and Thrall (1990) for a detailed discussion on these models.

Table 2.1. Multiplier Models

Frontier Type	Input-Oriented	Output-Oriented
	$\max \sum_{r=1}^{s}\mu_r y_{ro} + \mu$ subject to $\sum_{r=1}^{s}\mu_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} + \mu \le 0$ $\sum_{i=1}^{m} v_i x_{io} = 1$ $\mu_r, v_i \ge 0(\varepsilon)$	$\min \sum_{i=1}^{m} v_i x_{io} + v$ subject to $\sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s}\mu_r y_{rj} + v \ge 0$ $\sum_{r=1}^{s}\mu_r y_{ro} = 1$ $\mu_r, v_i \ge 0(\varepsilon)$
CRS	where $\mu = 0$	where $v = 0$
VRS	where μ free	where v free
NIRS	where $\mu \le 0$	where $v \ge 0$
NDRS	where $\mu \ge 0$	where $v \le 0$

Note that $\mu_r, v_i \geq \varepsilon$. This set of constraints ensures that a DMU with an efficiency score of one must be efficient. If a DMU's efficiency score equals one with non-zero slacks in an envelopment model, then this DMU must have a score less than one in the above related multiplier model (with ε). That is, if we impose $\mu_r, v_i \geq \varepsilon$ in the multiplier models, the two-stage process in the envelopment models is automatically carried out in the calculation. However, note that ε is a very small positive value and usually is set equal to 10^{-6}, and such choice does not always work. It is also possible that the multiplier model with can be infeasible because the ε is not correctly selected.

In the DEA literature, a number of approaches have been proposed to introduce additional restrictions on the values that the multipliers can assume.

Some of the techniques for enforcing these additional restrictions include imposing bounds on ratios of multipliers (Thompson et al., 1990), appending multiplier inequalities (Wong and Beasley, 1990), and requiring multipliers to belong to given closed cones (Charnes et al., 1989), among others.

We here present the assurance region (AR) approach of Thompson et al. (1990). To illustrate the AR approach, suppose we wish to incorporate additional inequality constraints of the following form into the multiplier DEA models as given in Table 2.1:

$$\begin{aligned} \alpha_i \leq \frac{v_i}{v_{i_o}} \leq \beta_i, \qquad i = 1,\ldots,m \\ \delta_r \leq \frac{\mu_r}{\mu_{r_o}} \leq \gamma_r, \qquad r = 1,\ldots,s \end{aligned} \qquad \text{(AR)}$$

Here, v_{i_o} and μ_{r_o} represent multipliers which serve as "numeraires" in establishing the upper and lower bounds represented here by α_i, β_i, and by δ_r, γ_r for the multipliers associated with inputs i =1, …, m and outputs r = 1, …, s where α_{i_o} = β_{i_o} = δ_{r_o} = γ_{r_o} = 1. The above constraints are called Assurance Region (AR) constraints as in Thompson et al. (1990).

Uses of such bounds are not restricted to prices. For example, Zhu (1996) uses an assurance region approach to establish bounds on the weights obtained from uses of Analytic Hierarchy Processes in Chinese textile manufacturing in order to reflect how the local government in measuring the textile manufacturing performance.

For example, we can include the following AR constraints

$$1 \leq \frac{v_{Employee}}{v_{Assets}} \leq 2.5$$

$$1.5 \le \frac{v_{Employee}}{v_{Equity}} \le 3$$

$$3 \le \frac{\mu_{MarketValue}}{\mu_{\mathrm{Re}venue}} \le 4$$

The first AR constraint indicates that Employee input should be at most 2.5 times as important as the Assets input, but at least as important as the Assets input.

It is noted that the AR constraints in the above form are non-linear, however, they can be converted into linear restrictions, namely

$$\alpha_i v_{i_o} \le v_i \le \beta_i v_{i_o}, \qquad i = 1,\ldots,m$$

$$\delta_r \mu_{r_o} \le \mu_r \le \gamma_r \mu_{r_o}, \qquad r = 1,\ldots,s$$

or

$$\alpha_i v_{i_o} \le v_i, \qquad i = 1,\ldots,m$$

$$v_i \le \beta_i v_{i_o}, \qquad i = 1,\ldots,m$$

$$\delta_r \mu_{r_o} \le \mu_r, \qquad r = 1,\ldots,s$$

$$\mu_r \le \gamma_r \mu_{r_o}, \qquad r = 1,\ldots,s$$

2.2 Multiplier Models in Spreadsheets

Figure 2.1 presents the input-oriented CRS multiplier spreadsheet model. We name the cells C2:E16 containing the inputs as "InputUsed" and the cells G2:H16 containing the outputs as "OutputProduced". Cells C19:E19 and G19:H19 are reserved for the decision variables – input and output multipliers, and are named "InputMultiplier" and "OutputMultiplier", respectively. Cells A2:A16 are reserved for DMU numbers which are used in the formulas in cells I2:I16.

Cell I2 contains the formula "= SUMPRODUCT(OutputMultiplier, INDEX (OutputProduced,A2,0))-SUMPRODUCT(InputMultiplier,INDEX (InputUsed,A2,0))" which represents the difference between weighted output and weighted input for DMU1. This value will be set as non-negative in the Solver parameters.

The function INDEX(array,row number,0) returns the entire row in the array. For example, the value for cell A2 is one, therefore INDEX(OutputProduced,A2,0) returns all the outputs for DMU1, i.e., cells G2:H2.

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	45840.8	
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	27508.7	
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	92123.8	
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-773759.6	
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	104598.3	
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	77833.6	
8	7	Ford Motor	243283	24547	346990		137137	4139	-473544	
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-188836.4	
10	9	Exxon	91296	40436	82000		110009	6470	-97253	
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-164259.7	
12	11	Wal-Mart	37871	14762	675000		93627	2740	-631266	
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-367744.2	
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-371061.1	
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-316571.1	
16	15	AT&T	88884	17274	299300		79609	139	-325710	
17									138870.6	
18										
19		**Multipliers**	1	1	1		1	1		
20		DMU under evaluation	**1**							
21		Efficiency	184711.4							
22										
23										
24										

Input multipliers changing cells

Output multipliers changing cells

Weighted inputs for DMU under evaluation

efficiency weighted output Target cell

reserved to indicate the DMU under evaluation

Figure 2.1. Input-oriented CRS Multiplier Spreadsheet Model

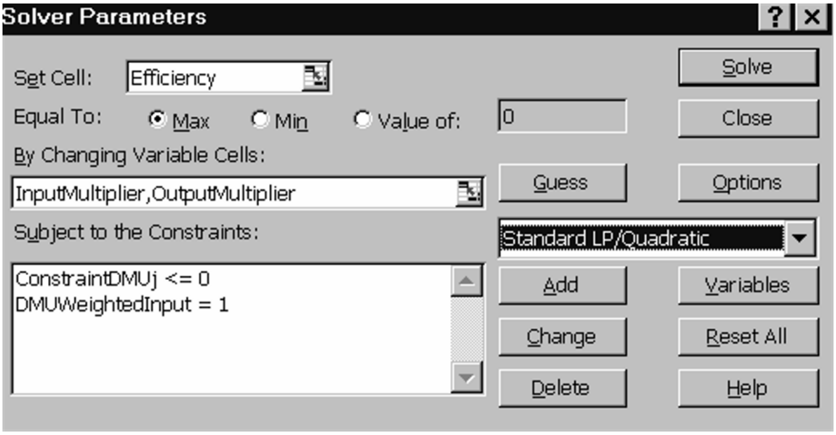


Figure 2.2. Premium Solver Parameters for Input-oriented CRS Multiplier Model

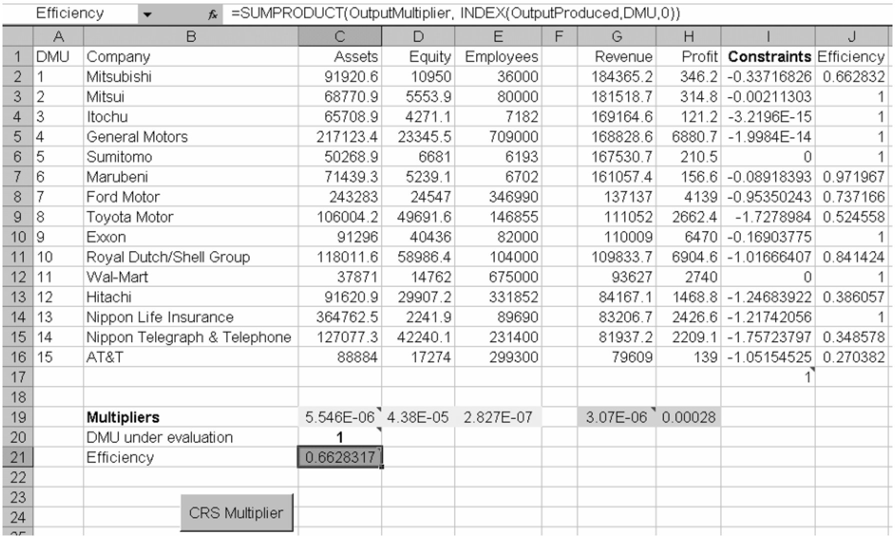
The formula in cell I2 is then copied into cells I3:I16. Cells I2:I16 are named “ConstraintDMUj”.

The formula for cell I17 is “= SUMPRODUCT (InputMultiplier, INDEX (InputUsed,DMU,0))”, where DMU is a range name for cell C20, indicating the DMU under evaluation. The value of cell I17 will be set equal to one in the Solver parameters. Cell I17 is named “DMUWeightedInput”.

The target cell is C21 which represents the efficiency – weighted output for the DMU under evaluation. The cell C21 is named “Efficiency”. Its formula is "= SUMPRODUCT(OutputMultiplier,INDEX(OutputProduced, DMU,0))".

Note that initial values of one are entered into the cells for the multipliers. As a result, some of the constraints are violated, and the value in cell C21 (efficiency) is greater than one. However, once the Solver solves, these values will be replaced by optimal solutions.

Figure 2.2 shows the Premium Solver parameters for the spreadsheet model in Figure 2.1. If one uses the Premium Solver, one should select "Standard LP/Quadratic" solver engine. In the Options, check the "Assume Non-Negative" box.



Efficiency | =SUMPRODUCT(OutputMultiplier, INDEX(OutputProduced,DMU,0))

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-0.33716826	0.662832
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	-0.00211303	1
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-3.2196E-15	1
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-1.9984E-14	1
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	0	1
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.08918393	0.971967
8	7	Ford Motor	243283	24547	346990		137137	4139	-0.95350243	0.737166
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-1.7278984	0.524558
10	9	Exxon	91296	40436	82000		110009	6470	-0.16903775	1
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-1.01666407	0.841424
12	11	Wal-Mart	37871	14762	675000		93627	2740	0	1
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-1.24683922	0.386057
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-1.21742056	1
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-1.75723797	0.348578
16	15	AT&T	88884	17274	299300		79609	139	-1.05154525	0.270382
17									1	
18										
19		**Multipliers**	5.546E-06	4.38E-05	2.827E-07		3.07E-06	0.00028		
20		DMU under evaluation	1							
21		Efficiency	0.6628317							
22										
23										
24		CRS Multiplier								

Figure 2.3. Input-oriented CRS Multiplier Efficiency

Figure 2.3 shows the optimal solutions for DMU1 with an efficiency of 0.66283. To calculate the CRS efficiencies for the remaining DMUs, we insert a VBA procedure "MultiplierCRS" to automate the computation, as shown in Figure 2.4. Note that the name of the module is changed to "MultiplierDEA". This VBA procedure works for other sets of DMUs when setting the "NDMUs", "NInputs", and "NOutputs" equal to proper values. In the current example, this VBA procedure takes the efficiency in cell C21 and places it into cells J2:J16, and also takes the optimal multipliers and places them into cells K2:M16 and O2:P16 for 15 DMUs. Select and run the macro "MultiplierCRS" in the Run Macro dialog box will generate the efficiency results. You may also create a button in Forms toolbar and assign macro "MultiplierCRS" to the button (see file "multiplier spreadsheet.xls" in the CD).

```
Sub MultiplierCRS()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
   NDMUs = 15
   NInputs = 3
   NOutputs = 2
   Dim i As Integer
   For i = 1 To NDMUs
   Range("DMU") = i
   SolverSolve UserFinish:=True
'record the efficiency scores
   Range("A1").Offset(i, NInputs + NOutputs + 4) = Range("Efficiency")
'record the optimal multipliers
   Range("InputMultiplier").Copy
   Range("A1").Offset(i, NInputs + NOutputs + 5).Select
   Selection.PasteSpecial Paste:=xlPasteValues
   Range("OutputMultiplier").Copy
   Range("A1").Offset(i, 2 * NInputs + NOutputs + 6).Select
   Selection.PasteSpecial Paste:=xlPasteValues
   Next i
End Sub
```

Figure 2.4. VBA Code for Input-oriented CRS Multiplier Model

Spreadsheets for other multiplier models can be set up in a similar manner. For example, Figure 2.5 shows a spreadsheet model for the input-oriented VRS multiplier model.

Because we have a decision variable that is free in sign, we need to introduce two variables in cells I19 and J19. The free variable in the VRS multiplier model is represented by cell J18 with a formula of "=I19-J19". In the Solver parameters, cells I19 and J19 (not cell J18) along with cells C19:E19 and G19:H19 are changing cells.

The formula for cell I2 is

Cell I2 =SUMPRODUCT(G2:H2,$G$19:$H$19)-SUMPRODUCT(C2:E2,$C$19:$E$19)+$I$19-$J$19

Cells for the multipliers and free variables are used as absolute references indicated by the dollar sign. This allows us to copy the formula in cell I2 to cells I3:I16. Figure 2.6 shows the Solver parameters for the input-oriented VRS multiplier spreadsheet model.

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-7.9936E-15	
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	0	
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-0.42009487	
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-4.3521E-14	
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	-0.2847084	
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.7938799	
8	7	Ford Motor	243283	24547	346990		137137	4139	-2.19408313	
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-2.22252322	
10	9	Exxon	91296	40436	82000		110009	6470	0	
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-0.098082	
12	11	Wal-Mart	37871	14762	675000		93627	2740	-3.39289316	
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-4.22139918	
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	-5.96194306	
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-4.06208857	
16	15	AT&T	88884	17274	299300		79609	139	-5.00032337	
17									1	
18									free variable	-6.81693
19		**Multipliers**	1.001E-05	0	2.217E-06		4.14E-05	0.00052	0	6.816927
20		DMU under evaluation	**1**							
21		Efficiency	1							

Figure 2.5.. Input-oriented VRS Multiplier Spreadsheet Model

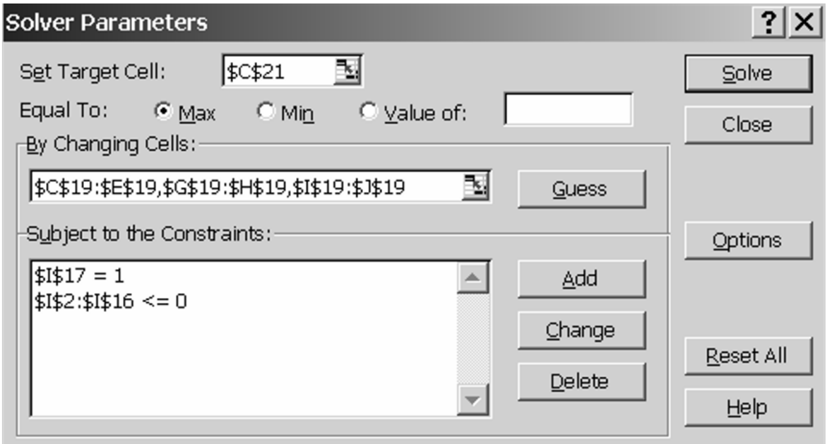
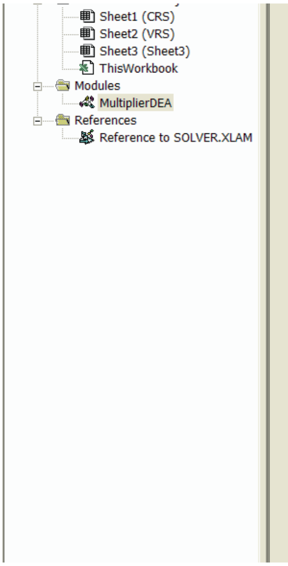


*Figure 2.6.*Solver Parameters for Input-oriented CRS Multiplier Model

Insert the VBA procedure "MultiplierVRS" shown in Figure 2.7 into the existing module "MultiplierDEA". The macro records the efficiency score in cells J2:J16, optimal free variable in cells K2:K16, and optimal multipliers in cells L2:N16 and P2:Q16 for 15 DMUs (see file "multiplier spreadsheet.xls" in the CD).




```
Sub MultiplierVRS()
'Declare i as integer. This i represents the DMU under
'evaluation. In the example, i goes form 1 to 15
    Dim i As Integer
    For i = 1 To 15
'set the value of cell C20 equal to i (1, 2,..., 15)

    Range("C20") = i
'Run the Solver model. The UserFinish is set to True so that
'the Solver Results dialog box will not be shown

    SolverSolve UserFinish:=True
'Place the efficiency in cell C20 into column J

    Range("J" & i + 1) = Range("C21")
'Select the cells containing the optimal multipliers

    Range("C19:H19").Select
'copy the selected multipliers and paste them to row "i+1"
'(that is row 2, 3, ..., 16) starting with column L

    Selection.Copy
    Range("L" & i + 1).Select
    Selection.PasteSpecial Paste:=xlPasteValues
'copy the value of free variable in cell J18 into column K

    Range("K" & i + 1) = Range("J18")
    Next
End Sub
```

Figure 2.7. VBA Code for the Input-oriented VRS Multiplier Model

	A	B	C	D	E	F	G	H	I	J
1	DMU	Company	Assets	Equity	Employees		Revenue	Profit	**Constraints**	Efficiency
2	1	Mitsubishi	91920.6	10950	36000		184365.2	346.2	-0.21073304	0.604129
3	2	Mitsui	68770.9	5553.9	80000		181518.7	314.8	-0.07467892	0.827172
4	3	Itochu	65708.9	4271.1	7182		169164.6	121.2	-2.3315E-15	1
5	4	General Motors	217123.4	23345.5	709000		168828.6	6880.7	-0.1888734	0.908288
6	5	Sumitomo	50268.9	6681	6193		167530.7	210.5	-0.04003305	1
7	6	Marubeni	71439.3	5239.1	6702		161057.4	156.6	-0.03864332	0.88263
8	7	Ford Motor	243283	24547	346990		137137	4139	-0.4430373	0.697857
9	8	Toyota Motor	106004.2	49691.6	146855		111052	2662.4	-1.18017405	0.372625
10	9	Exxon	91296	40436	82000		110009	6470	0	1
11	10	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6	-0.551944	0.836479
12	11	Wal-Mart	37871	14762	675000		93627	2740	-0.62040258	0.523164
13	12	Hitachi	91620.9	29907.2	331852		84167.1	1468.8	-1.03607407	0.280553
14	13	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6	5.5511E-15	1
15	14	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1	-1.19701102	0.317256
16	15	AT&T	88884	17274	299300		79609	139	-0.87923554	0.120764
17									1	
18										
19		**Multipliers**	1.172E-06	3.16E-05	1.172E-06		1.15E-06	0.00021		
20		DMU under evaluation	**15**							
21		Efficiency	0.1207645			AR				
22						1st	1.17E-06	<=	1.1715E-06	
23		CRS AR				2nd	1.17E-06	<=	2.9288E-06	
24										

Figure 2.8. CRS AR Multiplier Model

We next incorporate $1 \le \frac{v_{Employee}}{v_{Assets}} \le 2.5$ into the CRS multiplier model shown in Figure 2.3. The following two additional constraints are needed

$$1v_{Assets} \le v_{Employee} \text{ and } v_{Employee} \le 2.5v_{Assets}$$

Cells G22:G23 contains the left-hand-side of the above two constraints and cells I22:I23 contains the right-hand-side of the above two constraints, as shown in Figure 2.8. In the Solver parameters, we need to add these two additional constraints, as shown in Figure 2.9.

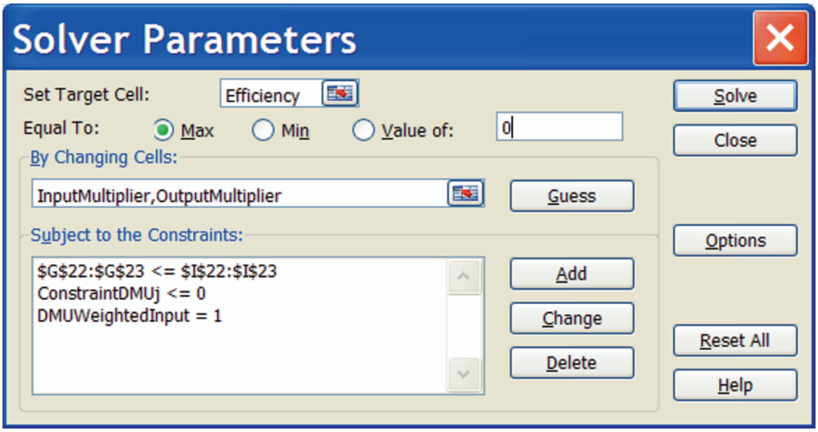


Figure 2.9. Solver Parameters for CRS AR Model

2.3 Slack-based Model

The input-oriented DEA models consider the possible (proportional) input reductions while maintaining the current levels of outputs. The output-oriented DEA models consider the possible (proportional) output augmentations while keeping the current levels of inputs. Charnes, Cooper, Golany, Seiford and Stutz (1985) develop an additive DEA model which considers possible input decreases as well as output increases simultaneously. The additive model is based upon input and output slacks. For example,

$$
\begin{aligned}
&\max \sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned}
\qquad (2.1)
$$

Note that model (2.1) assumes equal marginal worth for the nonzero input and output slacks. Therefore, caution should be excised in selecting the units for different input and output measures. Some *a priori* information may be required to prevent an inappropriate summation of non-commensurable measures. Previous management experience and expert opinion, which prove important in productivity analysis, may be used (see, e.g., Seiford and Zhu (1998)).

Model (2.1) therefore is modified to a weighted CRS slack-based model as follows (Ali, Lerme and Seiford, 1995).

$$\begin{aligned}
&\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+ \\
&\text{subject to} \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io} \qquad i = 1,2,\ldots,m; \\
&\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro} \qquad r = 1,2,\ldots,s; \\
&\lambda_j, s_i^-, s_r^+ \geq 0
\end{aligned} \tag{2.2}$$

where w_i^- and w_r^+ are user-specified weights obtained through value judgment. The DMU_o under evaluation will be termed efficient *if and only if* the optimal value to (2.2) is equal to zero. Otherwise, the nonzero optimal s_i^{-*} identifies an excess utilization of the *i*th input, and the non-zero optimal s_r^{+*} identifies a deficit in the *r*th output. Thus, the solution of (2.2) yields the information on possible adjustments to individual outputs and inputs of each DMU. Obviously, model (2.2) is useful for setting targets for inefficient DMUs with *a priori* information on the adjustments of outputs and inputs.

Table 2.2. Slack-based Models

Frontier type	Slack-based DEA Model
CRS	$\max \sum_{i=1}^{m} w_i^- s_i^- + \sum_{r=1}^{s} w_r^+ s_r^+$ subject to $\sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{io}$ $i = 1,2,\ldots,m;$ $\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{ro}$ $r = 1,2,\ldots,s;$ $\lambda_j, s_i^-, s_r^+ \geq 0$
VRS	Add $\sum_{j=1}^{n} \lambda_j = 1$
NIRS	Add $\sum_{j=1}^{n} \lambda_j \leq 1$
NDRS	Add $\sum_{j=1}^{n} \lambda_j \geq 1$

One should note that model (2.2) does not necessarily yield results that are different from those obtained from the model (2.1). In particular, it will not change the classification from efficient to inefficient (or vice versa) for any DMU.

Model (2.1) identifies a CRS frontier, and therefore is called CRS slack-based model. Table 2.2 summarizes the slack-based models in terms of the frontier types.

2.4 Slack-based Models in Spreadsheets

Figure 2.10 shows a spreadsheet model for the CRS slack-based model when DMU1 is under evaluation. Cells I2:I16 are reserved for λ_j. Cells F20:F24 are reserved for input and output slacks. The weights on slacks are entered into Cells G20:G24. Currently, the weights are all equal to one.

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees		Revenue	Profit		λ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		0
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0.2901
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		1.4476
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0
8	Ford Motor	243283	24547	346990		137137	4139		0
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0
10	Exxon	91296	40436	82000		110009	6470		0.001
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0
12	Wal-Mart	37871	14762	675000		93627	2740		0
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0
16	AT&T	88884	17274	299300		79609	139		0
17									
18		Reference		DMU under	1				
19	**Constraints**	set		Evaluation		**Slack**	**Weights**		
20	Assets	91920.6	=	91920.6		0	1		
21	Equity	10950	=	10950		0	1		
22	Employees	36000	=	36000		24871.42	1		
23	Revenue	184365.2	=	184365.2		107334.9	1		
24	Profit	346.2	=	346.2		0	1		
25						132206.4			

Figure 2.10. CRS Slack-based DEA Spreadsheet Model

Cells B20:B24 contain the following formulas

Cell B20 =SUMPRODUCT(B2:B16,$I$2:$I$16)+F20
Cell B21 =SUMPRODUCT(C2:C16,$I$2:$I$16)+F21
Cell B22 =SUMPRODUCT(D2:D16,$I$2:$I$16)+F22
Cell B23 =SUMPRODUCT(F2:F16,$I$2:$I$16)-F23
Cell B24 =SUMPRODUCT(G2:G16,$I$2:$I$16)-F24

The input and output values of the DMU under evaluation are placed into cells D20:D24 via the following formulas

Cell D20 =INDEX(B2:B16,E18,1)
Cell D21 =INDEX(C2:C16,E18,1)
Cell D22 =INDEX(D2:D16,E18,1)
Cell D23 =INDEX(F2:F16,E18,1)
Cell D24 =INDEX(G2:G16,E18,1)

Cell F25 is the target cell which represents the weighted slack. The formula for cell F25 is

Cell F25 =SUMPRODUCT(F20:F24,G20:G24)

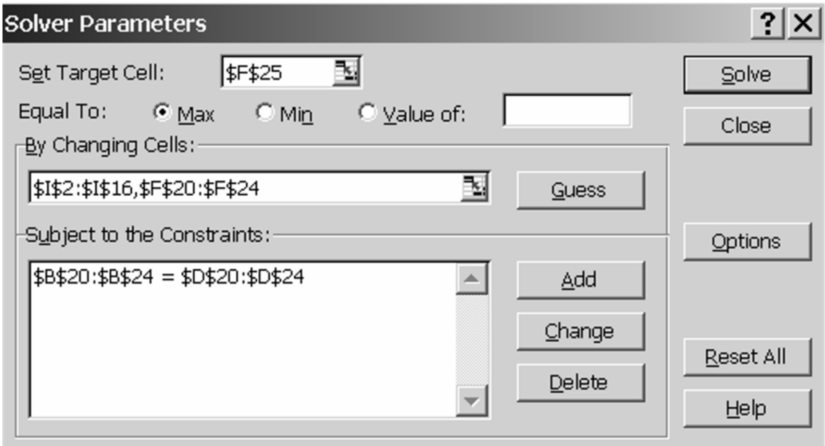


Figure 2.11. Solver Parameters for CRS Slack-based Model

Figure 2.11 shows the Solver parameters. Figure 2.12 shows the optimal slack values when DMU1 is under evaluation. Next, we insert a VBA procedure “CRSSlack” to calculate the optimal slacks for the remaining DMUs.

```
Sub CRSSlack()
Dim i As Integer
    For i = 1 To 15
'set the value of cell E18 equal to i (=1, 2,..., 15)
    Range("E18") = i
'Run the Slack Solver model
    SolverSolve UserFinish:=True
'Select the cells containing the slacks
    Range("F20:F24").Select
'record optimal slacks in cells K2:O16
```

```
Selection.Copy
Range("K" & i + 1).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next
End Sub
```

	J	K	L	M	N	O
1	Company	Assets	Equity	Employees	Revenue	Profit
2	Mitsubishi	0	0	24871.423	107334.9	0
3	Mitsui	0	0	-3.988E-10	0	0
4	Itochu	3.69E-11	0	0	-8.6E-14	1.75E-13
5	General Motors	0	0	-1.649E-10	0	0
6	Sumitomo	0	2.27E-12	1.99E-13	-2.1E-13	-7.1E-14
7	Marubeni	12794.2	0	0	5857.514	0.812974
8	Ford Motor	0	0	267229.26	112833	0
9	Toyota Motor	0	25289.03	108011.55	171436.8	0
10	Exxon	0	8.49E-12	2.118E-11	5.85E-11	0
11	Royal Dutch/Shell Group	0	13499.36	14957.88	78912.18	0
12	Wal-Mart	0	0	0	0	0
13	Hitachi	0	12685.39	307952.81	186551.3	0
14	Nippon Life Insurance	-1E-10	0	0	-1.9E-10	0
15	Nippon Telegraph & Telephone	0	17554.49	196254.33	288061.8	0
16	AT&T	0	5460.851	288349.72	216613.9	233.1999
17						
18						
19						
20						
21	CRS Slack					

Figure 2.12. CRS Slacks

	A	B	C	D	E	F	G	H	I
1	Company	Assets	Equity	Employees		Revenue	Profit		λ
2	Mitsubishi	91920.6	10950	36000		184365.2	346.2		1
3	Mitsui	68770.9	5553.9	80000		181518.7	314.8		0
4	Itochu	65708.9	4271.1	7182		169164.6	121.2		0
5	General Motors	217123.4	23345.5	709000		168828.6	6880.7		0
6	Sumitomo	50268.9	6681	6193		167530.7	210.5		0
7	Marubeni	71439.3	5239.1	6702		161057.4	156.6		0
8	Ford Motor	243283	24547	346990		137137	4139		0
9	Toyota Motor	106004.2	49691.6	146855		111052	2662.4		0
10	Exxon	91296	40436	82000		110009	6470		0
11	Royal Dutch/Shell Group	118011.6	58986.4	104000		109833.7	6904.6		0
12	Wal-Mart	37871	14762	675000		93627	2740		0
13	Hitachi	91620.9	29907.2	331852		84167.1	1468.8		0
14	Nippon Life Insurance	364762.5	2241.9	89690		83206.7	2426.6		0
15	Nippon Telegraph & Telephone	127077.3	42240.1	231400		81937.2	2209.1		0
16	AT&T	88884	17274	299300		79609	139		0
17									
18		Reference		DMU under	1				
19	**Constraints**	set		Evaluation		**Slack**	**Weights**		
20	Assets	91920.6	=	91920.6		0	1		
21	Equity	10950	=	10950		0	1		
22	Employees	36000	=	36000		0	1		
23	Revenue	184365.2	=	184365.2		0	1		
24	Profit	346.2	=	346.2		0	1		
25	Σλ	1	=	1		0			

Figure 2.13. VRS Slack-based Spreadsheet Model

By adding an additional constraint on $\sum_{j=1}^{n} \lambda_j$, we can obtain spreadsheet models for other slack-based models (see Excel file slack-based spreadsheet.xls in the CD). For example, Figure 2.13 shows a spreadsheet model for the VRS slack-based DEA model.

Range names are used in Figure 2.13. Cells B2:D16 are named "InputUsed" and cells F2:G16 are named "OutputProduced". We also name cells I2:I16 "Lambdas", cells F20:F24 "Slacks", G20:G24 "Weights", and cell E18 "DMU". Accordingly, we have formulas

Cell B20 = SUMPRODUCT(INDEX(InputUsed,0,1),Lambdas)+Slacks
Cell B21 = SUMPRODUCT(INDEX(InputUsed,0,2),Lambdas)+Slacks
Cell B22 = SUMPRODUCT(INDEX(InputUsed,0,3),Lambdas)+Slacks
Cell B23 = SUMPRODUCT(INDEX(OutputProduced,0,1),Lambdas)-Slacks
Cell B24 = SUMPRODUCT(INDEX(OutputProduced,0,2),Lambdas)-Slacks
Cell B25 = SUM(Lambdas)
Cell F25 = SUMPRODUCT(Slacks,Weights)

We then name cells B20:B24 "ReferenceSet", cells D20:D24 "DMUEvaluation", B25 "SumLambda", and cell F25 "SumSlack". Figure 2.14 shows the Solver parameters for the VRS slack-based model.

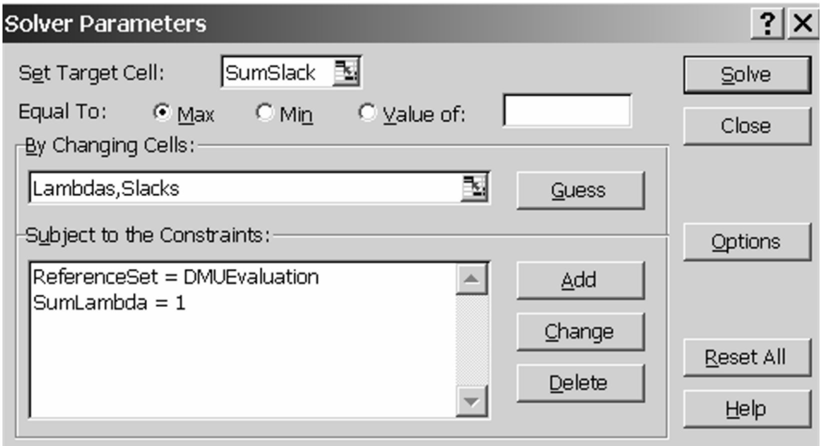


Figure 2.14. Solver Parameters for VRS Slack-based Model

Since range names are used in the Solver model, we can modify "CRSSlack" into a VBA procedure that can be applied to other data sets. The modified VBA procedure is called "Slack".

```
Sub Slack()
Dim NDMUs As Integer, NInputs As Integer, NOutputs As Integer
    NDMUs = 15
    NInputs = 3
    NOutputs = 2
Dim i As Integer
For i = 1 To NDMUs
Range("DMU") = i
SolverSolve UserFinish:=True
Range("Slacks").Copy
Range("A1").Offset(i, NInputs + NOutputs + 5).Select
Selection.PasteSpecial Paste:=xlPasteValues, Transpose:=True
Next
End Sub
```

2.5 Solving DEA Using DEAFrontier Software

2.5.1 Multiplier Model

To run the multiplier models, select the "Multiplier Model with Epsilon" menu item. You will be prompted with a form for selecting the models presented in Table 2.1. As shown in Figure 2.15, the default ε value = 0. The user can specify its own non-zero ε. The results are reported in a sheet named "Efficiency Report".

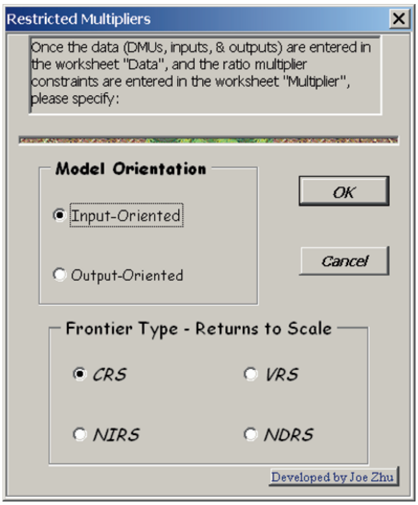


Figure 2.15. Multiplier Model

2.5.2 Restricted Multiplier Model

We need to set up the sheet "Multiplier" which contains the ARs. For example, if we want to include the following ARs

$$1 \le \frac{v_{Employee}}{v_{Assets}} \le 2.5$$

$$1.5 \le \frac{v_{Employee}}{v_{Equity}} \le 3$$

$$3 \le \frac{\mu_{MarketValue}}{\mu_{\mathrm{Re}venue}} \le 4$$

then the data in the "Multiplier" sheet should be entered as shown in the following Figure 2.16.

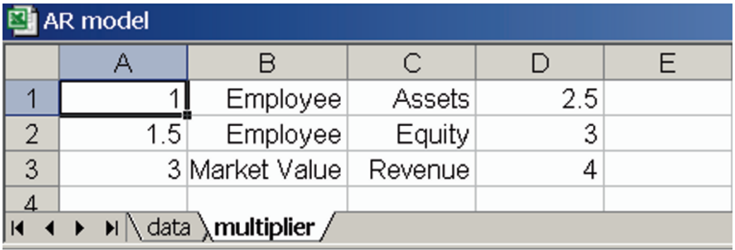


Figure 2.16. Restrictions (AR) on Multipliers

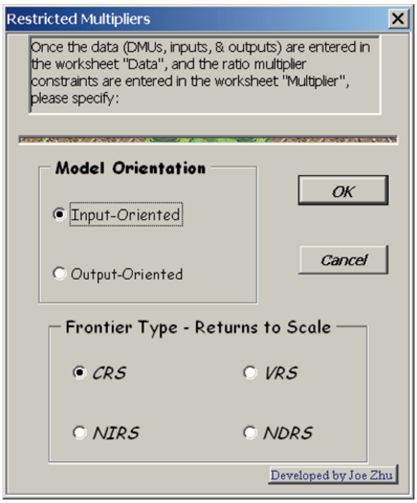
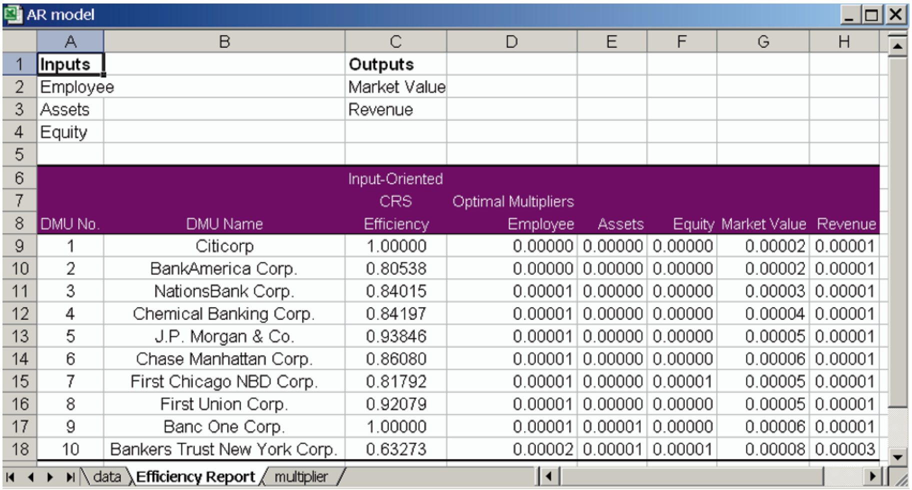


Figure 2.17. Restricted Multiplier Model

To avoid any errors, we suggest copying and pasting the input and output names from the "data" sheet when you enter the information into the "Multiplier" sheet. If the input (output) names in the two sheets do not match, the program will stop.

Once the "Multiplier" sheet is set up, select the "Restricted Multipliers" menu item and you will be prompted to choose a DEA model, as shown in Figure 2.17. Figure 2.18 shows the results of the input-oriented CRS multiplier model with the above ARs.

Note that you can also add ARs that link the input and output multipliers for the "Restricted Multipliers". Note also that if the ARs are not properly specified, then the related DEA model may be infeasible. If that happens, the program will return a value "-9999" for the efficiency score.


AR model

	A	B	C	D	E	F	G	H
1	Inputs		Outputs					
2	Employee		Market Value					
3	Assets		Revenue					
4	Equity							
5								
6			Input-Oriented					
7			CRS	Optimal Multipliers				
8	DMU No.	DMU Name	Efficiency	Employee	Assets	Equity	Market Value	Revenue
9	1	Citicorp	1.00000	0.00000	0.00000	0.00000	0.00002	0.00001
10	2	BankAmerica Corp.	0.80538	0.00000	0.00000	0.00000	0.00002	0.00001
11	3	NationsBank Corp.	0.84015	0.00001	0.00000	0.00000	0.00003	0.00001
12	4	Chemical Banking Corp.	0.84197	0.00001	0.00000	0.00000	0.00004	0.00001
13	5	J.P. Morgan & Co.	0.93846	0.00001	0.00000	0.00000	0.00005	0.00001
14	6	Chase Manhattan Corp.	0.86080	0.00001	0.00000	0.00000	0.00006	0.00001
15	7	First Chicago NBD Corp.	0.81792	0.00001	0.00000	0.00001	0.00005	0.00001
16	8	First Union Corp.	0.92079	0.00001	0.00000	0.00000	0.00005	0.00001
17	9	Banc One Corp.	1.00000	0.00001	0.00001	0.00000	0.00006	0.00001
18	10	Bankers Trust New York Corp.	0.63273	0.00002	0.00001	0.00001	0.00008	0.00003

data | Efficiency Report | multiplier

Figure 2.18. Restricted Multiplier Model Results

2.5.3 Slack-based Model

To run the slack-based models, select the "Slack-based Model" menu item. You will be prompted with a form for selecting the models presented in Table 2.2, as shown in Figure 2.19.

If you select "Yes" under the "Weights on Slacks", you will be asked to provide the weights, as shown in Figure 2.10. If you select "No", then all the weights are set equal to one.

The results are reported in a sheet named "Slack Report" along with a sheet named "Efficient Target".

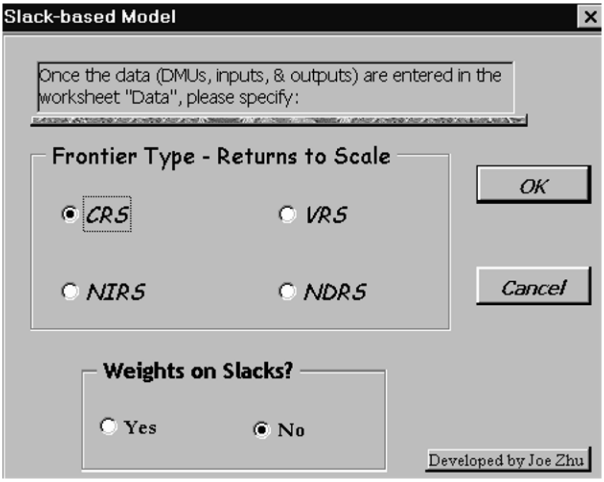


Figure 2.19. Slack-based Models

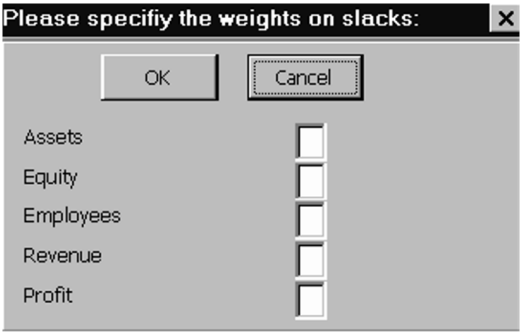


Figure 2.20. Weights on Slacks

REFERENCES

1. Ali, A. I., C.S. Lerme and L.M. Seiford (1995), Components of efficiency evaluation in data envelopment analysis, *European Journal of Operational Research*, 80, 462-473.
2. Charnes, A., W.W. Cooper, B. Golany, L.M. Seiford and J. Stutz (1985), Foundations of data envelopment analysis for Pareto-Koopman's efficient empirical production functions, *J. of Econometrics,* 30, 1-17.
3. Seiford, L.M. and R.M. Thrall (1990), Recent developments in DEA: the mathematical programming approach to frontier analysis, *Journal of Econometrics*, 46, 7-38.
4. Seiford, L.M. and J. Zhu (1998), Identifying excesses and deficits in Chinese industrial productivity (1953-1990): a weighted data envelopment analysis approach, *OMEGA*, 26, No. 2, 279-269.
5. Thompson, R. G., L. N. Langemeier, C. T. Lee, E. Lee, R. M. Thrall (1990), The role of multiplier bounds in efficiency analysis with application to Kansas farming. *J. Econometrics* 46, 93–108.
6. Wong, Y-H. B., and J. E. Beasley (1990), Restricting weight flexibility in data envelopment analysis, *Journal of Operational Research Society* 41/9, 829-835.
7. Zhu, J. (1996), Data envelopment analysis with preference structure, *Journal of Operational Research Society*, 47, 136-150.