Chapter 7

Benchmarking Models

7.1 Introduction

Gap analysis is often used as a fundamental method in performance evaluation and benchmarking. However, gap analysis only deals one measure at a time. It is rare that one single measure can suffice for the purpose of performance evaluation (Camp, 1995). As a result, some multi-factor based gap analysis methods have been developed. e.g., Spider charts, AHP maturity index, and Z charts. Although gaps can be identified with respect to individual performance measures, it remains a challenging task to combine the multiple measures in the final stage. Therefore, benchmarking models that can deal with multiple performance measures and provide an integrated benchmarking measure are needed.

Benchmarking is a process of defining valid measures of performance comparison among peer DMUs, using them to determine the relative positions of the peer DMUs and, ultimately, establishing a standard of excellence. In that sense, DEA can be regarded as a benchmarking tool, because the frontier identified can be regarded as an empirical standard of excellence.

Once the frontier is established, we may compare a set of new DMUs to the frontier. However, when a new DMU outperforms the identified frontier, a new frontier is generated by DEA. As a result, we do not have the same benchmark (frontier) for other (new) DMUs.

In the current chapter, we present a number of DEA-based benchmarking models where each (new) DMU is evaluated against a set of given benchmarks (standards).

7.2 Variable-benchmark Model

Cook, Seiford and Zhu (2004) develop a set of variable-benchmark model. Let E^* represent the set of benchmarks or the best-practice identified by the DEA. Based upon the input-oriented CRS envelopment model, we have

$$\begin{aligned}
&\min \delta^{CRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le \delta^{CRS} x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge y_r^{new} \\
&\lambda_j \ge 0, j \in E^*
\end{aligned} \tag{7.1}$$

where a new observation is represented by DMU^{new} with inputs x_i^{new} (i = 1, …, m) and outputs y_r^{new} (r = 1, …, s). The superscript of CRS indicates that the benchmark frontier composed by benchmark DMUs in set E^* exhibits CRS.

Model (7.1) measures the performance of DMU^{new} with respect to benchmark DMUs in set E^* when outputs are fixed at their current levels. Similarly, based upon the output-oriented CRS envelopment model, we can have a model that measures the performance of DMU^{new} in terms of outputs when inputs are fixed at their current levels.

$$\begin{aligned}
&\max \tau^{CRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge \tau^{CRS} y_r^{new} \\
&\lambda_j \ge 0, j \in E^*
\end{aligned} \tag{7.2}$$

Theorem 7.1 $\delta^{CRS*} = 1/\tau^{CRS*}$, where δ^{CRS*} is the optimal value to model (7.1) and τ_o^{CRS*} is the optimal value to model (7.2).

[Proof]: Suppose λ_j^* ($j \in E^*$) is an optimal solution associated with δ^{CRS*} in model (7.1). Now, let $\tau^{CRS*} = 1/\delta^{CRS*}$, and $\lambda_j' = \lambda_j^* \delta_o^{CRS*}$. Then τ^{CRS*} and λ_j' are optimal in model (7.2). Thus, $\delta^{CRS*} = 1/\tau^{CRS*}$. ■

Model (7.1) or (7.2) yields a benchmark for DMU^{new}. The ith input and the rth output for the benchmark can be expressed as

$$\begin{cases} \sum_{j \in E^*} \lambda_j^* x_{ij} & (ith \quad input) \\ \sum_{j \in E^*} \lambda_j^* y_{ij} & (rth \quad output) \end{cases} \tag{7.3}$$

Note also that although the DMUs associated with set E^* are given, the resulting benchmark may be different for each new DMU under evaluation. Because for each new DMU under evaluation, (7.3) may represent a different combination of DMUs associated with set E^*. Thus, models (7.1) and (7.2) represent a variable-benchmark scenario.

Theorem 7.2
(i) $\delta^{CRS^*} < 1$ or $\tau^{CRS^*} > 1$ indicates that the performance of DMU_o^{new} is dominated by the benchmark in (7.3).
(ii) $\delta^{CRS^*} = 1$ or $\tau^{CRS^*} = 1$ indicates that DMU^{new} achieve the same performance level of the benchmark in (7.3).
(iii) $\delta^{CRS^*} > 1$ or $\tau^{CRS^*} < 1$ indicates that input savings or output surpluses exist in DMU_o^{new} when compared to the benchmark in (7.3).

[Proof]: (i) and (ii) are obvious results in terms of DEA efficiency concept.

Now, $\delta^{CRS^*} > 1$ indicates that DMU^{new} can increase its inputs to reach the benchmark. This in turn indicates that δ^{CRS^*} - 1 measures the input saving achieved by DMU^{new}. Similarly, $\tau^{CRS^*} < 1$ indicates that DMU^{new} can decrease its outputs to reach the benchmark. This in turn indicates that 1 - τ^{CRS^*} measures the output surplus achieved by DMU^{new}. ■

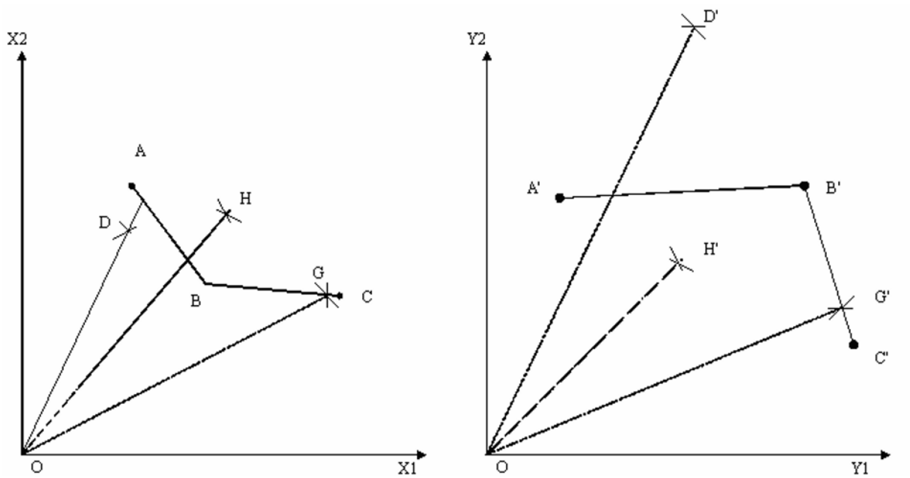


Figure 7.1. Variable-benchmark Model

Figure 7.1 illustrates the three cases described in Theorem 7.2. ABC (A'B'C') represents the input (output) benchmark frontier. D, H and G (or D', H', and G') represent the new DMUs to be benchmarked against ABC (or A'B'C'). We have $\delta_D^{CRS*} > 1$ for DMU D ($\tau_{D'}^{CRS*} < 1$ for DMU D') indicating that DMU D can increase its input values by δ_D^{CRS*} while producing the same amount of outputs generated by the benchmark (DMU D' can decrease its output levels while using the same amount of input levels consumed by the benchmark). Thus, $\delta_D^{CRS*} > 1$ is a measure of input savings achieved by DMU D and $\tau_{D'}^{CRS*} < 1$ is a measure of output surpluses achieved by DMU D'.

For DMU G and DMU G', we have $\delta_G^{CRS*} = 1$ and $\tau_{G'}^{CRS*} = 1$ indicating that they achieve the same performance level of the benchmark and no input savings or output surpluses exist. For DMU H and DMU H', we have $\delta_H^{CRS*} < 1$ and $\tau_{H'}^{CRS*} > 1$ indicating that inefficiency exists in the performance of these two DMUs.

Note that for example, in Figure 7.1, a convex combination of DMU A and DMU B is used as the benchmark for DMU D while a convex combination of DMU B and DMU C is used as the benchmark for DMU G. Thus, models (7.1) and (7.2) are called variable-benchmark models.

From Theorem 7.2, we can define δ^{CRS*} - 1 or 1 - τ^{CRS*} as the performance gap between DMU^{new} and the benchmark. Based upon δ^{CRS*} or τ^{CRS*}, a ranking of the benchmarking performance can be obtained.

It is likely that scale inefficiency may be allowed in the benchmarking. We therefore modify models (7.1) and (7.2) to incorporate scale inefficiency by assuming VRS.

$$\begin{aligned}
&\min \delta^{VRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le \delta^{VRS} x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge y_r^{new} \\
&\sum_{j \in E^*} \lambda_j = 1 \\
&\lambda_j \ge 0, j \in E^*
\end{aligned} \tag{7.4}$$

$$\begin{aligned}
&\max \tau^{VRS} \\
&\text{subject to} \\
&\sum_{j \in E^*} \lambda_j x_{ij} \le x_i^{new} \\
&\sum_{j \in E^*} \lambda_j y_{rj} \ge \tau^{VRS} y_r^{new} \\
&\sum_{j \in E^*} \lambda_j = 1 \\
&\lambda_j \ge 0, j \in E^*
\end{aligned} \tag{7.5}$$

Similar to Theorem 7.2, we have

Theorem 7.3
(i) $\delta^{VRS*} < 1$ or $\tau^{VRS*} > 1$ indicates that the performance of DMU^{new} is dominated by the benchmark in (7.3).
(ii) $\delta^{VRS*} = 1$ or $\tau^{VRS*} = 1$ indicates that DMU^{new} achieve the same performance level of the benchmark in (7.3).
(iii) $\delta^{VRS*} > 1$ or $\tau^{VRS*} < 1$ indicates that input savings or output surpluses exist in DMU^{new} when compared to the benchmark in (7.3).

Note that model (7.2) is always feasible, and model (7.1) is infeasible only if certain patterns of zero data are present (Zhu 1996b). Thus, if we assume that all the data are positive, (7.1) is always feasible. However, unlike models (7.1) and (7.2), models (7.4) and (7.5) may be infeasible.

Theorem 7.4
(i) If model (7.4) is infeasible, then the output vector of DMU^{new} dominates the output vector of the benchmark in (7.3).
(ii) If model (7.5) is infeasible, then the input vector of DMU^{new} dominates the input vector of the benchmark in (7.3).

[Proof]: The proof follows directly from the necessary and sufficient conditions for infeasibility in super-efficiency DEA model provided in Seiford and Zhu (1999). ■

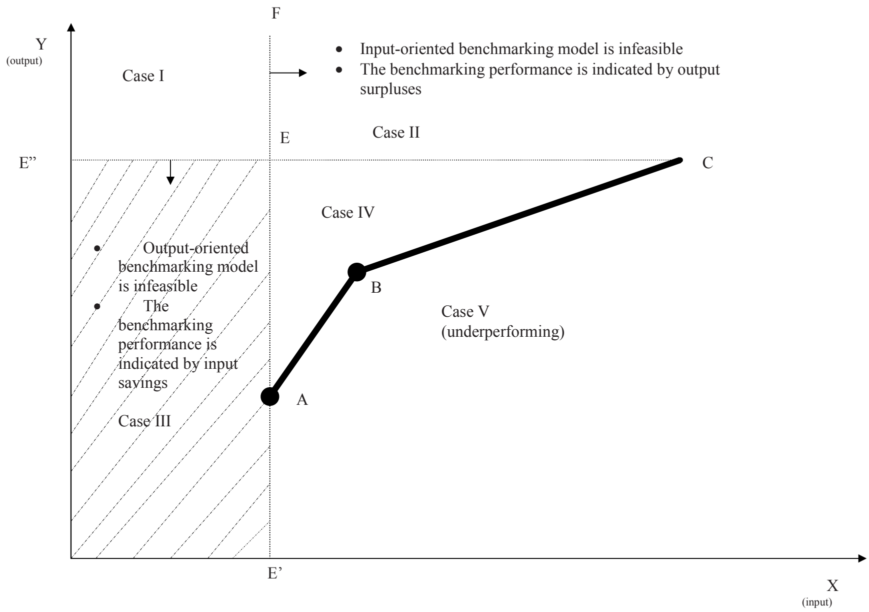


Figure 7.2. Infeasibility of VRS Variable-benchmark Model

The implication of the infeasibility associated with models (7.4) and (7.5) needs to be carefully examined. Consider Figure 7.2 where ABC represents the benchmark frontier. Models (7.4) and (7.5) yield finite optimal values for any DMU^{new} located below EC and to the right of EA. Model (7.4) is infeasible for DMU^{new} located above ray E"C and model (7.5) is infeasible for DMU^{new} located to the left of ray E'E.

Both models (7.4) and (7.5) are infeasible for DMU^{new} located above E"E and to the left of ray EF. Note that if DMU^{new} is located above E"C, its output value is greater than the output value of any convex combinations of A, B and C.

Note also that if DMU^{new} is located to the left of E'F, its input value is less than the input value of any convex combinations of A, B and C.

Based upon Theorem 7.4 and Figure 7.2, we have four cases:

Case I: When both models (7.4) and (7.5) are infeasible, this indicates that DMU^{new} has the smallest input level and the largest output level compared to the benchmark. Thus, both input savings and output surpluses exist in DMU^{new}.

Case II: When model (7.4) is infeasible and model (7.5) is feasible, the infeasibility of model (7.4) is caused by the fact that DMU^{new} has the largest output level compared to the benchmark. Thus, we use model (7.5) to characterize the output surpluses.

Case III: When model (7.5) is infeasible and model (7.4) is feasible, the infeasibility of model (7.5) is caused by the fact that DMU^{new} has the smallest input level compared to the benchmark. Thus, we use model (7.4) to characterize the input savings.

Case IV: When both models (7.4) and (7.5) are feasible, we use both of them to determine whether input savings and output surpluses exist.

If we change the constraint $\sum\lambda_j = 1$ to $\sum\lambda_j \leq 1$ and $\sum\lambda_j \geq 1$, then we obtain the NIRS and NDRS variable-benchmark models, respectively. Infeasibility may be associated with these two types of RTS frontiers, and we should apply the four cases discussed above. Table 7.1 summarizes the variable-benchmark models.

We next use 22 internet companies to illustrate the variable-benchmark models. Table 7.2 presents the data. We have four inputs: (1) number of website visitors (thousand), (2) number of employees (person), (3) marketing expenditure ($ million), and (4) development expenditure ($ million), and two outputs: (1) number of customers, and (2) revenue ($ million).

Table 7.1. Variable-benchmark Models

Frontier Type	Input-Oriented	Output-Oriented
CRS	$\min \delta^{Frontier}$ subject to $\sum_{j \in E^*} \lambda_j x_{ij} \leq \delta^{Frontier} x_i^{new}$ $\sum_{j \in E^*} \lambda_j y_{rj} \geq y_r^{new}$ $\lambda_j \geq 0, j \in E^*$	$\max \tau^{Frontier}$ subject to $\sum_{j \in E^*} \lambda_j x_{ij} \leq x_i^{new}$ $\sum_{j \in E^*} \lambda_j y_{rj} \geq \tau^{Frontier} y_r^{new}$ $\lambda_j \geq 0, j \in E^*$
VRS	Add $\sum\lambda_j = 1$	
NIRS	Add $\sum\lambda_j \leq 1$	
NDRS	Add $\sum\lambda_j \geq 1$	

Table 7.2. Data for the Internet Companies

Company	Visitors	Employee	Marketing	Develop-ment	Customers	Revenue
Barnes&Noble	64812	1237	111.55	21.01	4700000	202.57
Amazon.com	177744	7600	413.2	159.7	16900000	1640
CDnow	79848	502	89.73	23.42	3260000	147.19
eBay	168384	300	95.96	23.79	10010000	224.7
1-800-Flowers	11940	2100	92.15	8.07	7800000	52.89
Buy.com	27372	255	71.3	7.84	1950000	596.9
FTD.com	11856	75	29.93	5.29	1800000	62.6
Autobytel.com	12000	225	44.18	14.26	2065000	40.3
Beyond.com	17076	250	81.35	10.39	2000000	117.28
eToys	13896	940	120.46	43.43	1900000	151.04
E*Trade	29532	2400	301.7	78.5	1551000	621.4
Garden.com	16344	290	16	4.8	1070000	8.2
Drugstore.com	19092	408	61.5	14.9	695000	34.8
Outpost.com	7716	164	41.67	7	627000	188.6
iPrint	42132	225	8.13	3.54	380000	3.26
Furniture.com	10668	213	33.949	6.685	260000	10.904
PlanetRX.com	17124	390	55.18	12.95	254000	8.99
NextCard	46836	365	24.65	22.05	220000	26.56
PetsMart.com	18564	72	33.47	2.43	180000	10.45
Peapod	2076	1020	7.17	3.54	111900	73.13
Webvan	1680	1000	11.75	15.24	47000	13.31
CarsDirect.com	15612	702	33.43	2.14	12885	98.56

Suppose we select the first seven companies (Barnes & Noble, Amazon.com, CDnow, eBay, 1-800-Flowers, Buy.com, and FTD.com) as the benchmarks. If we apply the output-oriented CRS envelopment model to the seven companies, the top three companies (Barnes & Noble, Amazon.com, and CDnow) are not on the best-practice frontier, and therefore can be excluded. However, if we include them in the benchmark set, the benchmarking results will not be affected. Because λ_j^* related to the three companies must be equal to zero.

The spreadsheet model of the variable-benchmark models is very similar to the context-dependent DEA spreadsheet model. In fact, the evaluation background now is the selected benchmarks. Figure 7.3 shows the spreadsheet model for the output-oriented CRS variable-benchmark model where the benchmarks (evaluation background) are entered in rows 2-8.

Cell F2 is reserved to indicate the DMU under benchmarking. Cell F4 is the target cell which represent the τ_o^{CRS} in model (7.2). Cells I2:I8 represent the λ_j for the benchmarks. Cell B9 contains the formula "=SUMPRODUCT (B2:B8,$I$2:$I$8)". This formula is then copied into cells C9:E9. Cell G9 contains the formula "=SUMPRODUCT(G2:G8,$I$2:$I$8)". This formula is then copied into cell H9.

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	1.653097978	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.272959184
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0
9	**Benchmarks**	7471.438776	69.6045918	19.46199	2.14		532270.41	162.929337	
10		≤	≤	≤	≤		≥	≥	Benchmarking
11	**DMU under evaluation**	15612	702	33.43	2.14		21300.167	162.929337	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	1.095779422
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	1.327240986
14	eToys	13896	940	120.46	43.43	Variable	1900000	151.04	1.600761668
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	1.036374356
16	Garden.com	16344	290	16	4.8		1070000	8.2	1.42713759
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	4.852307242
18	Outpost.com	7716	164	41.67	7		627000	188.6	0.890769639
19	iPrint	42132	225	8.13	3.54		380000	3.26	2.231776947
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	7.369719683
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	12.93451824
22	NextCard	46836	365	24.65	22.05		220000	26.56	5.917828607
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	5.549892175
24	Peapod	2076	1020	7.17	3.54		111900	73.13	0.619051561
25	Webvan	1680	1000	11.75	15.24		47000	13.31	2.732844609
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	1.653097978

Figure 7.3. Output-oriented CRS Variable-benchmark Spreadsheet Model

Cells B11:E11, and Cells G11:H11 contain the formulas for the DMU under benchmarking – the right-hand-side of model (7.2). The formula for B11 is "=INDEX(B12:B26,$F$2,1)", and is copied into cells C11:E11. The formula for cell G11 is "=$F$4*INDEX(G12:G26,$F$2,1)", and is copied into cell H11.

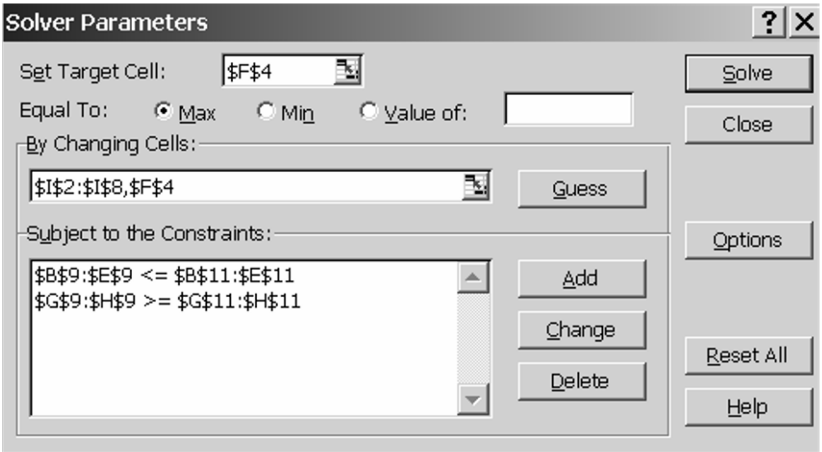


Figure 7.4. Solver Parameters for Output-oriented CRS Variable-benchmark Model

Figure 7.4 shows the Solver parameters for the spreadsheet model shown in Figure 7.3. A VBA procedure "VariableBenchmark" is used to record the benchmarking scores into cells I12:I26.

```
Sub VariableBenchmark()
Dim i As Integer
For i = 1 To 15
Range("F2") = i
SolverSolve UserFinish:=True
Range("I" & i + 11) = Range("F4")
Next
End Sub
```

Because the model in Figure 7.3 is an output-oriented model, a smaller score (τ^{CRS*}) indicates a better performance. Thus, Peapod is the best company with respective to the specified benchmarks. The non-zero optimal λ_j^* indicates the actual benchmark for a company under benchmarking. For example, Buy.com is used as the actual benchmark for CarsDirect.com (see cell I7 in Figure 7.3).

If we use the input-oriented CRS variable-benchmark model, we need change the formula for cell B11 in Figure 7.3 to "=$F$4*INDEX (B12:B26,$F$2,1)". This formula is then copied into cells C11:E11. The formula for cell G11 is changed to "=INDEX(G12:G26,$F$2,1)" and is copied into cell H11. All the other formulas in Figure 7.3 remain unchanged.

We also need to change the Solver parameters shown in Figure 7.4 by selecting "Min", as shown in Figure 7.5. Figure 7.6 shows the spreadsheet model for the input-oriented CRS variable-benchmark model and the benchmarking scores. It can be seen that Theorem 7.1 is true.

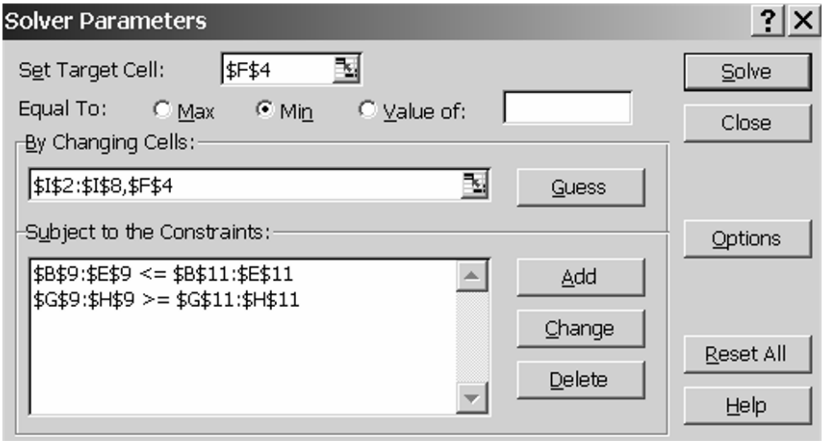


Figure 7.5. Solver Parameters for Input-oriented CRS Variable-benchmark Model

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	0.604924822	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.165119786
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0
9	**Benchmarks**	4519.65877	42.1055453	11.77304	1.294539119		321983.58	98.56	input-oriented
10		|∧	|∧	|∧	|∧		∨|	∨|	Benchmarking
11	**DMU under evaluation**	9444.086319	424.657225	20.22264	1.294539119		12885	98.56	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	0.912592425
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	0.753442676
14	eToys	13896	940	120.46	43.43	Variable	1900000	151.04	0.624702615
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	0.9649023
16	Garden.com	16344	290	16	4.8		1070000	8.2	0.700703287
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	0.206087527
18	Outpost.com	7716	164	41.67	7		627000	188.6	1.122624702
19	iPrint	42132	225	8.13	3.54		380000	3.26	0.448073452
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	0.135690371
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	0.077312505
22	NextCard	46836	365	24.65	22.05		220000	26.56	0.168980899
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	0.180183681
24	Peapod	2076	1020	7.17	3.54		111900	73.13	1.615374328
25	Webvan	1680	1000	11.75	15.24		47000	13.31	0.365919085
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	0.604924822

Figure 7.6. Input-oriented CRS Variable-benchmark Spreadsheet Model

We now consider the input-oriented VRS variable-benchmark model. We need to add a cell representing $\sum \lambda_j$ in the spreadsheet shown in Figure 7.6. We select cell I9, and enter the formula "=SUM(I2:I8)". We also need to add an additional constraint on $\sum \lambda_j = 1$ in the Solver parameters shown in Figure 7.5. This constraint is "$I$9 = 1", as shown in Figure 7.7.

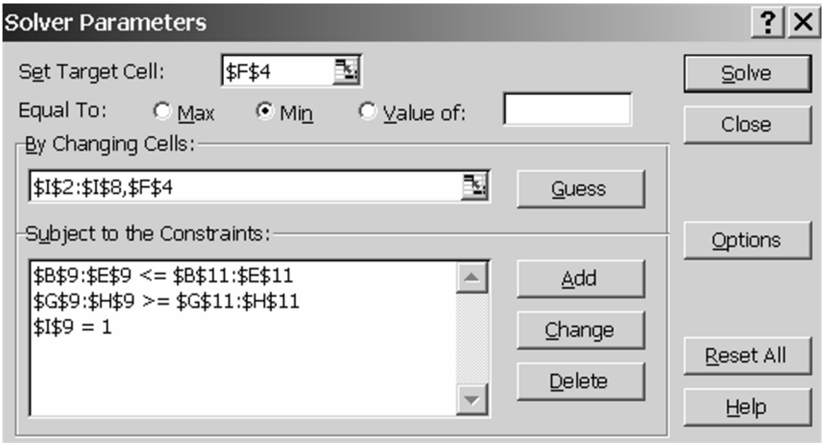


Figure 7.7. Solver Parameters for Input-oriented VRS Variable-benchmark Model

	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	λ
2	Barnes&Noble	64812	1237	111.55	21.01	15	4700000	202.57	0
3	Amazon.com	177744	7600	413.2	159.7	Score	16900000	1640	0
4	Cdnow	79848	502	89.73	23.42	2.552160133	3260000	147.19	0
5	eBay	168384	300	95.96	23.79		10010000	224.7	0
6	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	0
7	Buy.com	27372	255	71.3	7.84		1950000	596.9	0.067303013
8	FTD.com	11856	75	29.93	5.29		1800000	62.6	0.932696987
9	**Benchmarks**	12900.27355	87.1145424	32.71433	5.461622684		1810095.5	98.56	1
10		≤	≤	≤	≤		≥	≥	
11	**DMU under evaluation**	39844.32399	1791.61641	85.31871	5.461622684		12885	98.56	Score
12	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	0.988309167
13	Beyond.com	17076	250	81.35	10.39		2000000	117.28	0.787957227
14	eToys	13896	940	120.46	43.43	VRS Variable	1900000	151.04	1.038346917
15	E*Trade	29532	2400	301.7	78.5	Benchmark	1551000	621.4	0.95011279
16	Garden.com	16344	290	16	4.8		1070000	8.2	Infeasible
17	Drugstore.com	19092	408	61.5	14.9		695000	34.8	Infeasible
18	Outpost.com	7716	164	41.67	7		627000	188.6	Infeasible
19	iPrint	42132	225	8.13	3.54		380000	3.26	Infeasible
20	Furniture.com	10668	213	33.949	6.685		260000	10.904	Infeasible
21	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	Infeasible
22	NextCard	46836	365	24.65	22.05		220000	26.56	Infeasible
23	PetsMart.com	18564	72	33.47	2.43		180000	10.45	Infeasible
24	Peapod	2076	1020	7.17	3.54		111900	73.13	5.858280241
25	Webvan	1680	1000	11.75	15.24		47000	13.31	Infeasible
26	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	2.552160133

Figure 7.8. Input-oriented VRS Variable-benchmark Spreadsheet Model

Figure 7.8 shows the spreadsheet for the input-oriented VRS variable-benchmark model and the benchmarking scores in cells I12:I26. The button "VRS Variable Benchmark" is linked to the VBA procedure "VRSVariableBenchmark".

```
Sub VRSVariableBenchmark()
Dim i As Integer
For i = 1 To 15
```

```
Range("F2") = i
SolverSolve UserFinish:=True
If SolverSolve(UserFinish:=True) = 5 Then
Range("I" & i + 11) = "Infeasible"
Else
Range("I" & i + 11) = Range("F4")
End If
Next
End Sub
```

Because of the VRS frontier, the model may be infeasible. The SolverSolve function returns an integer value that indicates Solver's "success". If this value is 5, it means that there are no feasible solutions. This is represented by the statement "SolverSolve(UserFinish:=True) = 5". In the procedure, if the Solver returns a value of 5, then the procedure records "infeasible". Otherwise, the procedure records the optimal value in cell F4 of Figure 7.8.

7.3 Fixed-benchmark Model

Although the benchmark frontier is given in the variable-benchmark models, a DMU^{new} under benchmarking has the freedom to choose a subset of benchmarks so that the performance of DMU^{new} can be characterized in the most favorable light. Situations when the same benchmark should be fixed are likely to occur. For example, the management may indicate that DMUs A and B in Figure 7.1 should be used as the fixed benchmark. i.e., DMU C in Figure 7.1 may not be used in constructing the benchmark.

To couple with this situation, Cook, Seiford and Zhu (2004) turn to the multiplier models. For example, the input-oriented CRS multiplier model determines a set of referent best-practice DMUs represented by a set of binding constraints in optimality. Let set $\boldsymbol{B} = \{ DMU_j : j \in \mathbf{I}_B \}$ be the selected subset of benchmark set E^*. i.e., $\mathbf{I}_B \subset E^*$. Based upon the input-oriented CRS multiplier model, we have

$$
\begin{aligned}
&\tilde{\sigma}^{\text{CRS*}} = \max \sum_{r=1}^{s} \mu_r y_r^{new} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} = 0 \qquad j \in \mathbf{I}_B \\
&\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} \le 0 \qquad j \notin \mathbf{I}_B \\
&\sum_{i=1}^{m} v_i x_i^{new} = 1 \\
&\mu_r, v_i \ge 0.
\end{aligned}
\tag{7.6}
$$

By applying equalities in the constraints associated with benchmark DMUs, model (7.6) measures DMU^{new}'s performance against the benchmark constructed by set ***B***. At optimality, some DMU$_j$ $j \notin \mathbf{I}_B$, may join the fixed-benchmark set if the associated constraints are binding.

Note that model (7.6) may be infeasible. For example, the DMUs in set ***B*** may not be fit into the same facet when they number greater than m+s-1, where *m* is the number of inputs and *s* is the number of outputs. In this case, we need to adjust the set ***B***.

Three possible cases are associated with model (7.6). $\tilde{\sigma}^{CRS*}$ > 1 indicating that DMU^{new} outperforms the benchmark. $\tilde{\sigma}^{CRS*}$ = 1 indicating that DMU^{new} achieves the same performance level of the benchmark. $\tilde{\sigma}^{CRS*}$ < 1 indicating that the benchmark outperforms DMU^{new}.

By applying RTS frontier type and model orientation, we obtain the fixed-benchmark models in Table 7.3

Table 7.3. Fixed-benchmark Models

Frontier Type	Input-Oriented	Output-Oriented
	$\max \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} + \mu = 0 \quad j \in \mathbf{I}_B$ $\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} v_i x_{ij} + \mu \le 0 \quad j \notin \mathbf{I}_B$ $\sum_{i=1}^{m} v_i x_i^{new} = 1$ $\mu_r, v_i \ge 0$	$\min \sum_{i=1}^{m} v_i x_i^{new} + v$ subject to $\sum_{i=1}^{s} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v = 0 \quad j \in \mathbf{I}_B$ $\sum_{i=1}^{s} v_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + v \ge 0 \quad j \notin \mathbf{I}_B$ $\sum_{r=1}^{s} \mu_r y_r^{new} = 1$ $\mu_r, v_i \ge 0$
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 $v \le 0$

DMU^{new} is not included in the constraints of $\sum_{r=1}^{s} \mu_r y_{rj}$ - $\sum_{i=1}^{m} v_i x_{ij}$ $+ \mu \le 0$ ($j \notin \mathbf{I}_B$) ($\sum_{i=1}^{m} v_i x_{ij}$ - $\sum_{r=1}^{s} \mu_r y_{rj}$ $+ v \ge 0$ ($j \notin \mathbf{I}_B$)). However, other peer DMUs (($j \notin \mathbf{I}_B$) are included.

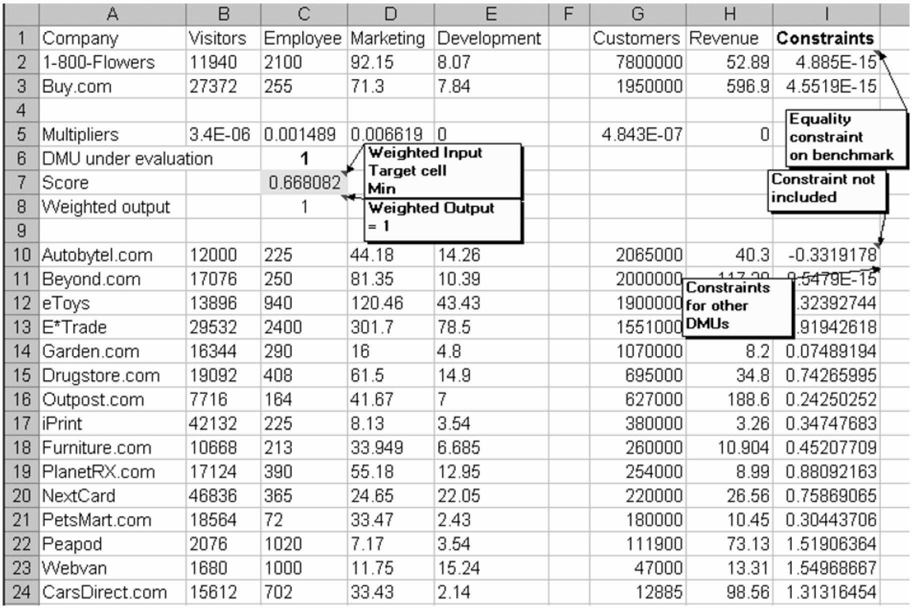
Figure 7.9 shows the output-oriented CRS fixed-benchmark spreadsheet model where 1-800-Flowers and Buy.com are two fixed benchmarks. Cells B5:E5 and G5:H5 are reserved for input and output multipliers, respectively. They are the changing cells in the Solver parameters.

Cell C7 is the target cell and contains the formula "=SUMPRODUCT (B5:E5,INDEX(B10:E24,C6,0))", where cell C6 indicates the DMU under evaluation – Autobytel.com.

Cell C8 contains the formula representing $\sum_{r=1}^{s} \mu_r y_r^{new}$

Cell C8=SUMPRODUCT(G5:H5,INDEX (G10:H24,C6,0))

The formula for cell I2 is "=SUMPRODUCT(B2:E2,$B$5:$E$5)-SUMPRODUCT(G2:H2,$G$5:$H$5)", and is copied into cells I3 and I10:I24.



	A	B	C	D	E	F	G	H	I
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	**Constraints**
2	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	4.885E-15
3	Buy.com	27372	255	71.3	7.84		1950000	596.9	4.5519E-15
4									
5	Multipliers	3.4E-06	0.001489	0.006619	0		4.843E-07	0	
6	DMU under evaluation		1						
7	Score		0.668082						
8	Weighted output		1						
9									
10	Autobytel.com	12000	225	44.18	14.26		2065000	40.3	-0.3319178
11	Beyond.com	17076	250	81.35	10.39		2000000	[illegible]	[illegible].5479E-15
12	eToys	13896	940	120.46	43.43		1900000	[illegible]	[illegible].32392744
13	E*Trade	29532	2400	301.7	78.5		1551000	[illegible]	[illegible].91942618
14	Garden.com	16344	290	16	4.8		1070000	8.2	0.07489194
15	Drugstore.com	19092	408	61.5	14.9		695000	34.8	0.74265995
16	Outpost.com	7716	164	41.67	7		627000	188.6	0.24250252
17	iPrint	42132	225	8.13	3.54		380000	3.26	0.34747683
18	Furniture.com	10668	213	33.949	6.685		260000	10.904	0.45207709
19	PlanetRX.com	17124	390	55.18	12.95		254000	8.99	0.88092163
20	NextCard	46836	365	24.65	22.05		220000	26.56	0.75869065
21	PetsMart.com	18564	72	33.47	2.43		180000	10.45	0.30443706
22	Peapod	2076	1020	7.17	3.54		111900	73.13	1.51906364
23	Webvan	1680	1000	11.75	15.24		47000	13.31	1.54968667
24	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	1.31316454

Figure 7.9. Output-oriented CRS Fixed-benchmark Spreadsheet Model

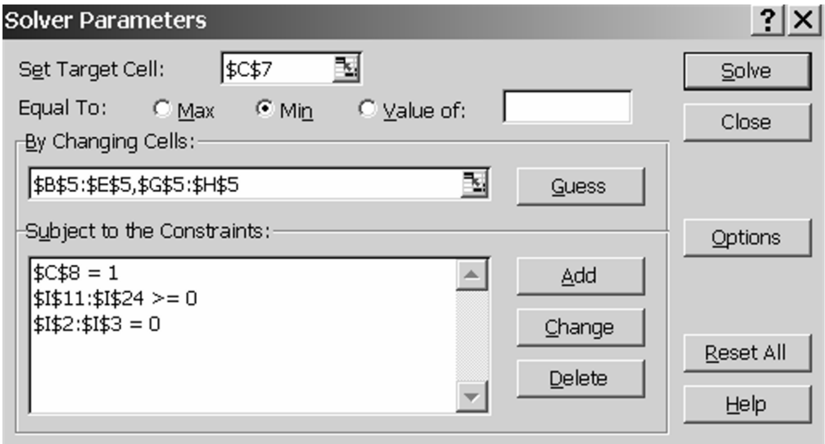


Figure 7.10. Solver Parameters for Output-oriented CRS Fixed-benchmark Model

Figure 7.10 shows the Solver parameters for Autobytel.com. Note that we have "$I$2:$I$3 = 0" for the two benchmarks. Note also that "$I$11:$I$24 >=0" does not include the DMU under evaluation, Autobytel.com.

To solve the remaining DMUs, we need to set up different Solver parameters. Because the constraints change for each DMU under evaluation. For example, if we change the value of cell C6 to 15, i.e., we benchmark CarsDirect.com, we obtain a set of new Solver parameters by removing "$I$24>=0" from the Solver parameters shown in Figure 7.10 and then adding "$I$10>=0", as shown in Figure 7.11.

Because different Solver parameters are used for different DMUs under benchmarking, a set of sophisticated VBA codes is required to automate the calculation. We here do not discuss it, and suggest using the "DEA Excel Solver" – a DEA Add-In for Microsoft Excel described in Chapter 12 to obtain the scores (see cells J10:J24 in Figure 7.11).

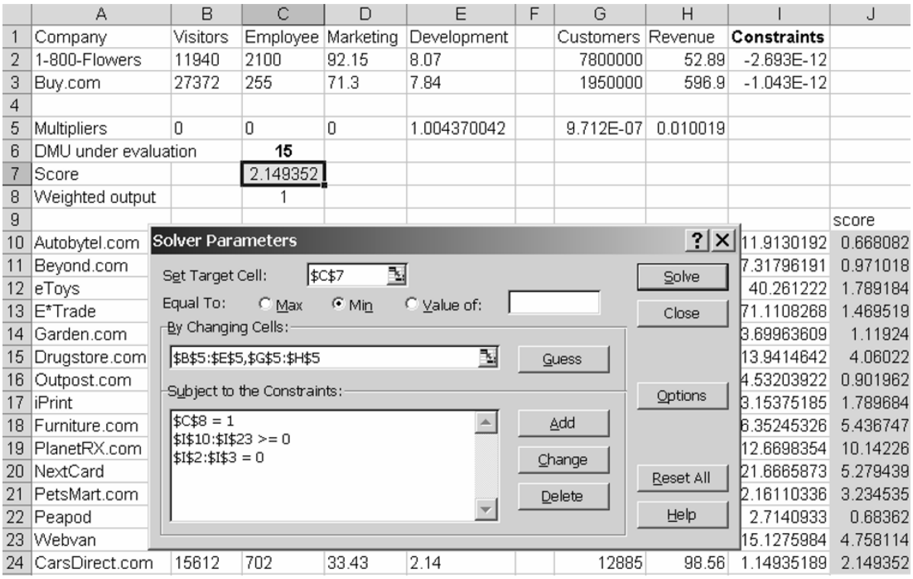


Figure 7.11. Output-oriented CRS Fixed-benchmark Scores for Internet Companies

7.4 Fixed-benchmark Model and Efficiency Ratio

A commonly used measure of efficiency is the ratio of output to input. For example, profit per employee measures the labor productivity. When multiple inputs and outputs are present, we may define the following efficiency ratio

$$\frac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}}$$

where v_i and u_r represent the input and output weights, respectively.

DEA calculate the ratio efficiency without the information on the weights. In fact, the multiplier DEA models can be transformed into linear fractional programming problems. For example, if we define $\nu_i = \text{t}\, v_i$ and $\mu_r = \text{t}\, u_r$, where t = $1/\sum \nu_i x_{io}$, the input-oriented CRS multiplier model can be transformed into

$$\begin{aligned} &\max \frac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}} \\ &\text{subject to} \\ &\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \quad j = 1,2,\ldots,n \\ &u_r, v_i \ge 0 \qquad \forall r,i \end{aligned} \tag{7.7}$$

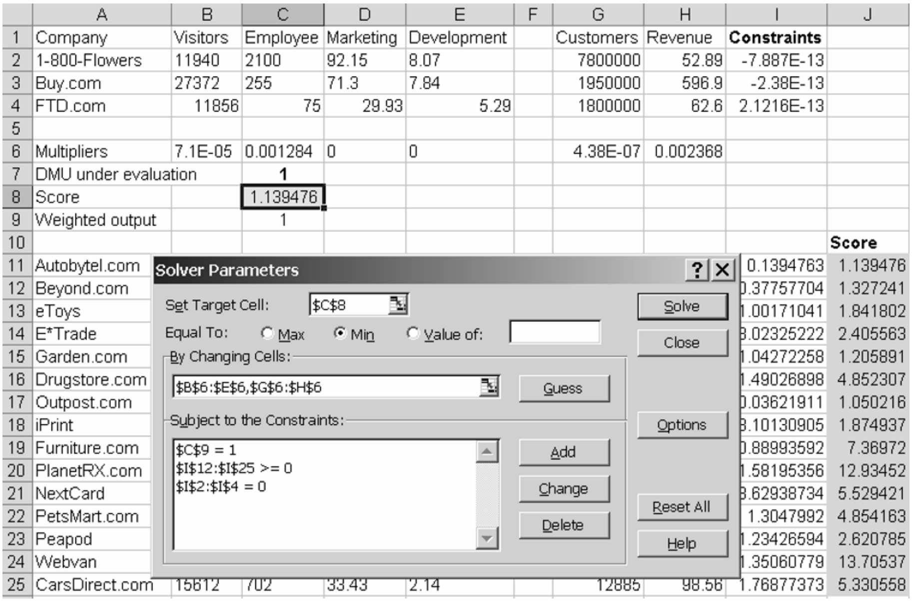
The objective function in (7.7) represents the efficiency ratio of a DMU under evaluation. Because of the constraints in (7.7), the (maximum) efficiency cannot exceed one. Consequently, a DMU with an efficiency score of one is on the frontier. It can be seen that no additional information on the weights or tradeoffs are incorporated into the model (7.7).

If we apply the input-oriented CRS fixed-benchmark model to (7.7), we obtain

$$\begin{aligned} &\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\ &\text{subject to} \\ &\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} = 1 \quad j \in \mathbf{I}_{\text{B}} \\ &\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1 \quad j \notin \mathbf{I}_{\text{B}} \\ &u_r, v_i \ge 0 \qquad \forall r,i \end{aligned} \tag{7.8}$$

It can be seen from (7.8) that the fixed benchmarks incorporate implicit tradeoff information into the efficiency evaluation. i.e., the constraints associated with $\mathbf{I}_B$ can be viewed as incorporation of tradeoffs or weight restrictions in DEA. Model (7.8) yields the (maximum) efficiency under the implicit tradeoff information represented by the benchmarks.

As more DMUs are selected as fixed benchmarks, more complete information on the weights becomes available. For example, if we add FTD.com to the fixed-benchmark set, the benchmarking score for Autobytel.com becomes 1.1395, as shown in Figure 7.12. As expected, the performance of those internet companies becomes worse when the set of fixed benchmarks expands.



	A	B	C	D	E	F	G	H	I	J
1	Company	Visitors	Employee	Marketing	Development		Customers	Revenue	**Constraints**	
2	1-800-Flowers	11940	2100	92.15	8.07		7800000	52.89	-7.887E-13	
3	Buy.com	27372	255	71.3	7.84		1950000	596.9	-2.38E-13	
4	FTD.com	11856	75	29.93	5.29		1800000	62.6	2.1216E-13	
5										
6	Multipliers	7.1E-05	0.001284	0	0		4.38E-07	0.002368		
7	DMU under evaluation		1							
8	Score		1.139476							
9	Weighted output		1							
10										**Score**
11	Autobytel.com								0.1394763	1.139476
12	Beyond.com								.37757704	1.327241
13	eToys								1.00171041	1.841802
14	E*Trade								3.02325222	2.405563
15	Garden.com								1.04272258	1.205891
16	Drugstore.com								1.49026898	4.852307
17	Outpost.com								.03621911	1.050216
18	iPrint								3.10130905	1.874937
19	Furniture.com								.88993592	7.36972
20	PlanetRX.com								1.58195356	12.93452
21	NextCard								3.62938734	5.529421
22	PetsMart.com								1.3047992	4.854163
23	Peapod								1.23426594	2.620785
24	Webvan								1.35060779	13.70537
25	CarsDirect.com	15612	702	33.43	2.14		12885	98.56	1.76877373	5.330558

Figure 7.12. Spreadsheet Model and Solver Parameters for Fixed-benchmark Model

Similarly, the output-oriented CRS fixed-benchmark model is equivalent to

$$\min \frac{\sum_{i=1}^{m} v_i x_i^{new}}{\sum_{r=1}^{s} u_r y_r^{new}}$$

subject to

$$\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} = 1 \quad j \in \mathbf{I}_B$$

$$\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} \geq 1 \quad j \notin \mathbf{I}_B$$
$$u_r, v_i \geq 0 \qquad \forall r, i$$

Note that we may define an ideal benchmark whose *r*th output y_r^{ideal} is the maximum output value across all DMUs, and *i*th input x_i^{ideal} the minimum input value across all DMUs. If we replace the fixed-benchmark set by the ideal benchmark, we have

$$\begin{aligned} &\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\ &\text{subject to} \\ &\frac{\sum_{r=1}^{s} u_r y_r^{ideal}}{\sum_{i=1}^{m} v_i x_i^{ideal}} = 1 \\ &\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1 \\ &u_r, v_i \geq 0 \qquad \forall r, i \end{aligned} \tag{7.9}$$

Because the ideal benchmark dominates all DMUs (unless DMU$_j$ is one of the ideal benchmark), the optimal value to (7.9) must not be greater than one. Further, $\sum u_r y_{rj} / \sum v_i x_{ij} \leq 1$ are redundant, and model (7.9) can be simplified as

$$\begin{aligned} &\max \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\ &\text{subject to} \\ &\frac{\sum_{r=1}^{s} u_r y_r^{ideal}}{\sum_{i=1}^{m} v_i x_i^{ideal}} = 1 \\ &u_r, v_i \geq 0 \qquad \forall r, i \end{aligned} \tag{7.10}$$

Model (7.10) is equivalent to the following linear programming problem

$$
\begin{aligned}
&\max \sum_{r=1}^{s} \mu_r y_r^{new} \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} v_i x_i^{ideal} = 0 \qquad (7.11) \\
&\sum_{i=1}^{m} v_i x_i^{new} = 1 \\
&\mu_r, v_i \geq 0.
\end{aligned}
$$

Model (7.10) or (7.11) calculate the maximum efficiency of a specific DMU under evaluation given that the efficiency of the ideal benchmark is set equal to one. If we introduce RTS frontier type and model orientation into (7.10), we obtain other ideal-benchmark models, as shown in Table 7.4.

Table 7.4. Ideal-benchmark Models

Frontier Type	Input-Oriented	Output-Oriented
	$\max \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} v_i x_i^{ideal} + \mu = 0$ $\sum_{i=1}^{m} v_i x_i^{new} = 1$ $\mu_r, v_i \geq 0.$	$\min \sum_{i=1}^{m} v_i x_i^{new} + v$ subject to $\sum_{i=1}^{s} v_i x_i^{ideal} - \sum_{r=1}^{s} \mu_r y_r^{ideal} + v = 0$ $\sum_{r=1}^{s} \mu_r y_r^{new} = 1$ $\mu_r, v_i \geq 0$
CRS	where $\mu = 0$	where $v = 0$
VRS	where μ free	where v free
NIRS	where $\mu \leq 0$	where $v \geq 0$
NDRS	where $\mu \geq 0$	where $v \leq 0$

7.5 Minimum Efficiency Model

Note that the fixed-benchmark models yield the maximum efficiency scores when the tradeoffs are implicitly defined by the benchmarks. If we change the objective function of model (7.8) into minimization, we have

$$
\begin{aligned}
&\min \frac{\sum_{r=1}^{s} u_r y_r^{new}}{\sum_{i=1}^{m} v_i x_i^{new}} \\
&\text{subject to} \qquad (7.12) \\
&\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} = 1 \quad j \in \mathbf{I}_{\mathrm{B}}
\end{aligned}
$$

$$\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1 \quad j \notin \mathbf{I}_{\mathrm{B}}$$
$$u_r, v_i \geq 0 \qquad \forall r, i$$

We refer to (7.12) as the input-oriented CRS minimum efficiency model. Although the benchmarks implicitly define the tradeoffs amongst inputs and outputs, the exact tradeoffs are still unavailable to us. Thus, the optimal value to (7.12) gives the lower efficiency bound for DMU^{new}. The optimal value to (7.8) yields the upper efficiency bound. The true efficiency of DMU^{new} lies in-between the bounds.

In fact, model (7.12) describes the worst efficiency scenario whereas model (7.8) describe the best efficiency scenario. The minimum efficiency for the original input-oriented DEA models (e.g., model (7.7)) is zero, and for the original output-oriented DEA models is infinite.

Similarly, we can obtain the output-oriented CRS minimum efficiency model,

$$\begin{aligned}
&\max \frac{\sum_{i=1}^{m} v_i x_i^{new}}{\sum_{r=1}^{s} u_r y_r^{new}} \\
&\text{subject to} \\
&\frac{\sum_{i=1}^{m} v_i x_{ij}}{\sum_{r=1}^{s} u_r y_{rj}} = 1 \quad j \in \mathbf{I}_{\mathrm{B}} \\
&u_r, v_i \geq 0 \qquad \forall r, i
\end{aligned} \tag{7.13}$$

Recall that a smaller score indicates a better performance in the output-oriented DEA models. Therefore, the output-oriented CRS minimum efficiency score (optimal value to model (7.13) is greater than or equal to the efficiency score obtained from the output-oriented CRS fixed-benchmark model.

The linear program equivalents to (7.12) and (7.13) are presented in Table 7.5 which summarizes the minimum efficiency models.

The spreadsheet models for the minimum efficiency models are similar to the fixed-benchmark spreadsheet models. We only need to change the "Max" to "Min" in the Solver parameters for the input-oriented models, and change the "Min" to "Max" for the output-oriented models. For example, consider the output-oriented CRS fixed-benchmark model shown in Figure 7.9. Figure 7.13 shows the corresponding minimum efficiency spreadsheet model.

Table 7.5. Minimum Efficiency Models

Frontier Type	Input-Oriented	Output-Oriented
	$\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$	$\max \sum_{i=1}^{m} \nu_i x_i^{new} + \nu$
	subject to	subject to
	$\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} \nu_i x_{ij} + \mu = 0 \quad j \in \mathbf{I}_B$	$\sum_{i=1}^{s} \nu_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + \nu = 0 \quad j \in \mathbf{I}_B$
	$\sum_{r=1}^{s} \mu_r y_{rj} - \sum_{i=1}^{s} \nu_i x_{ij} + \mu \le 0 \quad j \notin \mathbf{I}_B$	$\sum_{i=1}^{s} \nu_i x_{ij} - \sum_{r=1}^{s} \mu_r y_{rj} + \nu \ge 0 \quad j \notin \mathbf{I}_B$
	$\sum_{i=1}^{m} \nu_i x_i^{new} = 1$	$\sum_{r=1}^{s} \mu_r y_r^{new} = 1$
	$\mu_r, \nu_i \ge 0$	$\mu_r, \nu_i \ge 0$
CRS	where $\mu = 0$	where $\nu = 0$
VRS	where μ free	where ν free
NIRS	where $\mu \le 0$	where $\nu \ge 0$
NDRS	where $\mu \ge 0$	where $\nu \le 0$

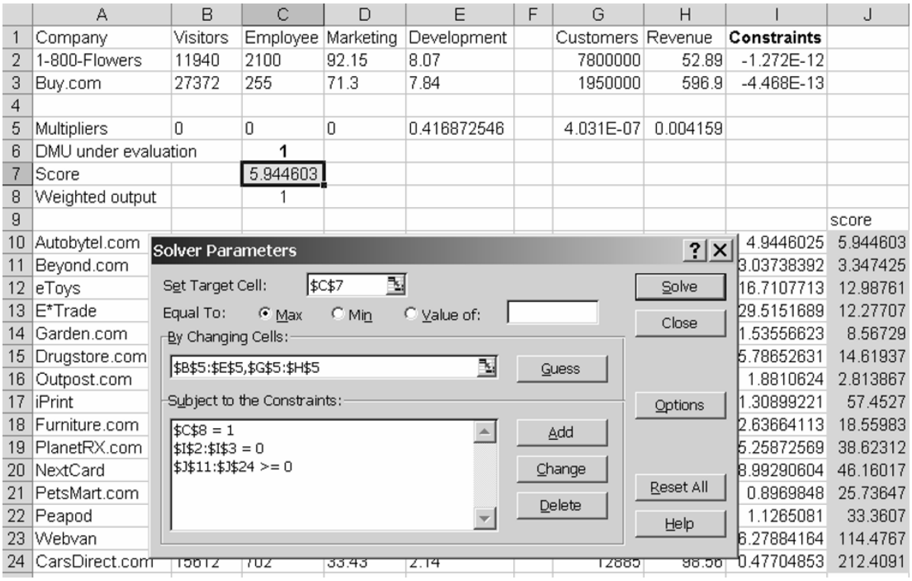


Figure 7.13. Output-oriented CRS Minimum Efficiency Spreadsheet Model

Under the tradeoffs characterized by the two benchmarks, the true efficiency of Autobytel.com lies in [0.6681, 5.9446]. Cells J10:J24 report the "minimum efficiency" for the 15 internet companies. The scores are calculated by the DEA Excel Solver discussed in Chapter 12.

If we introduce the ideal benchmark into the minimum efficiency models, we obtain, for example, the input-oriented VRS ideal-benchmark minimum efficiency model

$$\begin{aligned}
&\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu \\
&\text{subject to} \\
&\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} \nu_i x_i^{ideal} + \mu = 0 \\
&\sum_{i=1}^{m} \nu_i x_i^{new} = 1 \\
&\mu_r, \nu_i \geq 0 \text{ and } \mu \text{ free in sign}
\end{aligned} \tag{7.14}$$

Table 7.6 presents the ideal-benchmark minimum efficiency models.

Table 7.6. Ideal-benchmark Minimum Efficiency Models

Frontier Type	Input-Oriented	Output-Oriented
	$\min \sum_{r=1}^{s} \mu_r y_r^{new} + \mu$ subject to $\sum_{r=1}^{s} \mu_r y_r^{ideal} - \sum_{i=1}^{s} \nu_i x_i^{ideal} + \mu = 0$, $\sum_{i=1}^{m} \nu_i x_i^{new} = 1$, $\mu_r, \nu_i \geq 0.$	$\max \sum_{i=1}^{m} \nu_i x_i^{new} + \nu$ subject to $\sum_{i=1}^{s} \nu_i x_i^{ideal} - \sum_{r=1}^{s} \mu_r y_r^{ideal} + \nu = 0$, $\sum_{r=1}^{s} \mu_r y_r^{new} = 1$, $\mu_r, \nu_i \geq 0$
CRS	where $\mu = 0$	where $\nu = 0$
VRS	where μ free	where ν free
NIRS	where $\mu \leq 0$	where $\nu \geq 0$
NDRS	where $\mu \geq 0$	where $\nu \leq 0$

7.6 Buyer-seller Efficiency Model

As pointed out by Wise and Morrison (2000), one of the major flaws in the current business-to-business (B2B) model is that it focuses on price-driven transactions between buyers and sellers, and fails to recognize other important vendor attributes such as response time, quality and customization. In fact, a number of efficiency-based negotiation models have been developed to deal with multiple attributes – inputs and outputs. For example, DEA is used by Weber and Desai (1996) to develop models for vendor evaluation and negotiation. The fixed-benchmark models and the minimum efficiency models can better help the vendor in evaluating and selecting the vendors.

Talluri (2002) proposes a buyer-seller game model that evaluates the efficiency of alternative bids with respect to the ideal target set by the buyer. Zhu (2004) shows that this buyer-seller game model is closely related to DEA and can be simplified as the models presented in Tables 7.4 and 7.6.

We next use the data in Table 5.1 to demonstrate the use of DEA benchmarking models. A Fortune 500 pharmaceutical company was involved in the implementation of a Just-in-Time manufacturing system. Therefore, price, delivery performance, and quality were considered to be the three most important criteria in evaluating and selecting vendors. In Weber and Desai (1996), the price criterion is measured by the total purchase price based on a per unit contract delivered price, the delivery criterion is measured by the percentage of late deliveries, and the quality criterion is measured by the percentage of units rejected. Obviously, the measures for delivery and quality are bad outputs. Therefore, we re-define the delivery and quality by percentage on-time deliveries and percentage of accepted units, respectively. (Otherwise, we should use the method described in Chapter 5.)

Table 7.7. Data for the Six Vendors

Vendor	Price ($/unit)	% accepted units	% on-time deliveries
1	0.1958	98.8	95
2	0.1881	99.2	93
3	0.2204	100	100
4	0.2081	97.9	100
5	0.2118	97.7	97
6	0.2096	98.8	96

Table 7.8. Input-oriented CRS Efficiency and Efficient Target for Vendors

Vendor	Efficiency	Price ($/units)	% acceptance	% on-time deliveries
1	0.981	0.192145	**101.3333**	95
2	1	0.1881	99.2	93
3	0.918	0.202258	**106.6667**	100
4	0.972	0.202258	**106.6667**	100
5	0.926	0.19619	**103.4667**	97
6	0.926	0.194168	**102.4**	96

The results are based upon the input-oriented CRS envelopment model.

Table 7.7 presents the data for six vendors that are obtained from the data presented in Table 5.1. The second column reports the input, and the third and forth columns report the two outputs. We next need to determine the frontier type. Because the outputs are measured in percentages, we assume the vendors form a VRS frontier. Otherwise, unreasonable results may be obtained if we assume CRS frontier. For example, Table 7.8 reports the input-oriented CRS efficiency scores (second column) with the efficient targets. It can be seen that the efficient targets on percentage of accepted units are impossible to achieve.

If we use the input-oriented VRS envelopment model, vendors 2, 3, and 4 are efficient, and can be selected. However, if we specify an ideal benchmark by the minimum input value and the maximum output values, as shown in Figure 7.14, we can further characterize the six vendors.

	A	B	C	D	E	F	G
1		Price ($/units)		% accepted units	% on-time deliveries	constraint	free variable= F4-G4
2	Ideal target	0.1881		100	100	0	
3						free variable	0.897424
4	multipliers	4.770992		0	0	0.8974237	0
5	DMU under evaluation		6				
6	Score		0.8974				
7	Weighted input		1			Maximum	
8						Efficiency	
9	Vendor 1	0.1958		98.8	95	0.9606742	
10	Vendor 2	0.1881		99.2	93	1	
11	Vendor 3	0.2204		100	100	0.8534483	Max
12	Vendor 4	0.2081		97.9	100	0.9038924	
13	Vendor 5	0.2118		97.7	97	0.888102	
14	Vendor 6	0.2096		98.8	96	0.8974237	

Figure 7.14. Input-oriented VRS Ideal-benchmark Spreadsheet Model

Figure 7.14 shows the spreadsheet for the input-oriented VRS ideal-benchmark model. Cell C4 and cells D4:E4 are reserved for the input and output multipliers. The free variable is represented by cell G3 which contains the formula "=F4-G4". Cells F4:G4 are specified as changing cells in the Solver parameters (see Figure 7.15).

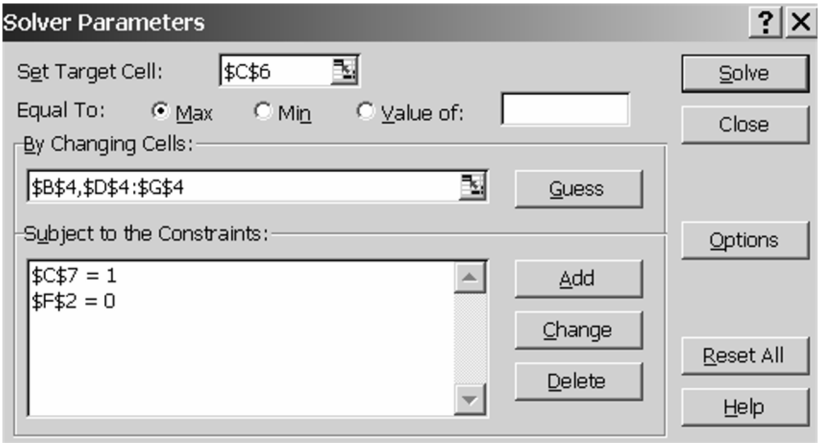


Figure 7.15. Solver Parameters for Input-oriented VRS Ideal-benchmark Model

Cell F2 contains the formula for the ideal benchmark, that is

Cell F2=SUMPRODUCT(D2:E2,D4:E4)-B2*B4+G3

Cell C5 is reserved to indicate the vendor under evaluation. The (maximum) efficiency is presented in cell C6 which contains the formula

Cell C6=SUMPRODUCT(D4:E4,INDEX(D9:E14,C5,0))+G3

Cell C7 is the weighted input and contains the formula

Cell C7=B4*INDEX(B9:B14,C5,1)

The Solver parameters shown in Figure 7.15 remain the same for all the vendors, and the calculation is performed by the VBA procedure "IdealBenchmark".

```
Sub IdealBenchmark()
Dim i As Integer
For i = 1 To 6
Range("C5") = i
SolverSolve UserFinish:=True
Range("F" & i + 8) = Range("C6")
Next
End Sub
```

Based upon the scores in cells F9:F14 in Figure 7.14, vendor 2 has the best performance.

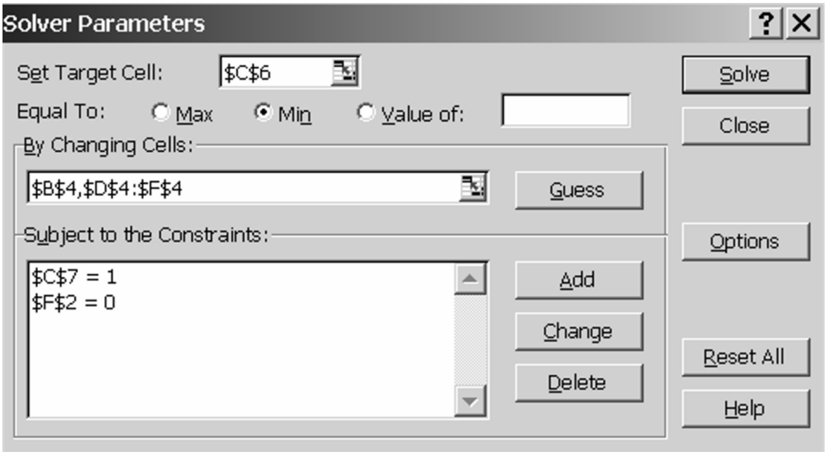


Figure 7.16. Solver Parameters for VRS Ideal-benchmark Minimum Efficiency Model

Next, we turn to the ideal-benchmark minimum efficiency model (7.14). The spreadsheet is the same as the one shown in Figure 7.14. However, we need to change “Max” to “Min” in the Solver parameters shown in Figure 7.15. Figure 7.16 shows the result. Figure 7.17 shows the minimum efficiency scores in cells F9:F14. The minimum efficiency model also indicates that vendor 2 is the best one.

	A	B	C	D	E	F	G
1		Price ($/units)		% accepted units	% on-time deliveries	constraint	
2	Ideal target	0.1881		100	100	-1.11E-16	
3						free variable	0
4	multipliers	4.770992		0	0.008974	0	0
5	DMU under evaluation		**6**				
6	Score		0.8615				
7	Weighted input		1			Minimum	
8						Efficiency	
9	Vendor 1	0.1958		98.8	95	0.9126404	
10	Vendor 2	0.1881		99.2	93	0.93	Min
11	Vendor 3	0.2204		100	100	0.8534483	
12	Vendor 4	0.2081		97.9	100	0.8849106	
13	Vendor 5	0.2118		97.7	97	0.8614589	
14	Vendor 6	0.2096		98.8	96	0.8615267	

Figure 7.17. Minimum Efficiency Scores for the Six Vendors

7.7 Solving DEA Using DEAFrontier Software

7.7.1 Variable-benchmark Models

To run the variable-benchmark models presented in Table 7.1, we need set up the data sheets. *Store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. The format for these two sheets is the same as that shown in Figure 12.3. Then select the Variable Benchmark Model menu item. You will be prompted a form for selecting the model orientation and the frontier type as shown in Figure 7.18. Note that if you select a frontier type other than CRS, the results may be infeasible. The benchmarking results are reported in the sheet "Benchmarking Results".

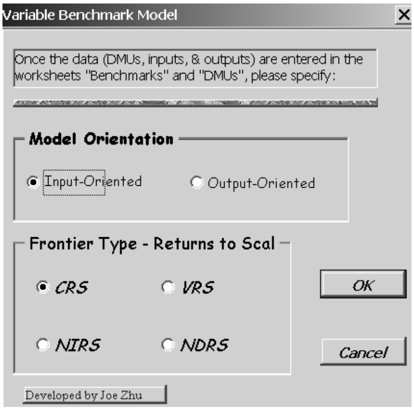


Figure 7.18. Variable Benchmark Models

7.7.2 Fixed-benchmark Models

To run the fixed-benchmark models presented in Table 7.3, we *store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. Then select the Fixed-Benchmark Model menu item. You will be prompted a form for selecting the model orientation and the frontier type. The results are reported in the "Efficiency Report" sheet. If the benchmarks are not properly selected, you will have infeasible results and need to adjust the benchmarks.

The Ideal-benchmark Models in Table 7.4 should be calculated using the Fixed-Benchmark Model menu item. The data for the ideal benchmark is stored in the "Benchmarks" sheet.

7.7.3 Minimum Efficiency Models

To run the minimum efficiency models presented in Table 7.5, we *store the benchmarks in a sheet named "Benchmarks" and the DMUs under evaluation in a sheet named "DMUs"*. Then select the Minimum Efficiency Model menu item. You will be prompted a form for selecting the model orientation and the frontier type. The results are reported in the "Minimum Efficiency" sheet.

The Ideal-benchmark Minimum Efficiency Models in Table 7.6 should be calculated using the Minimum Efficiency menu item. The data for the ideal benchmark is stored in the "Benchmarks" sheet.

REFERENCES

1. Camp, R.C., (1995), *Business Process Benchmarking, Finding and Implementing Best Practices*. ASQC Quality Press, Milwaukee, Wisconsin.
2. Cook, W.D., L.M. Seiford and J. Zhu (2004), Models for performance benchmarking: Measuring the effect of e-commerce activities on banking performance, *OMEGA*, Vol. 32, Issue 4, 313-322
3. Seiford, L.M. and J. Zhu (1999), Infeasibility of super efficiency data envelopment analysis models, *INFOR*, 37, No. 2, 174-187.
4. Talluri, S. (2002), A buyer–seller game model for selection and negotiation of purchasing bids. European Journal of Operational Research 143 (1), 171–180.
5. Weber, C.A. and A. Desai (1996), Determination of paths to vendor market efficiency using parallel coordinates representation: a negotiation tool for buyers, *European Journal of Operational Research*, 90, 142-155.
6. Zhu, J. (2004), A buyer-seller game model for selection and negotiation of purchasing bids: Extensions and New Models, *European Journal of Operational Research,* Vol. 154, Issue 1, 150-156.