MaxDEA

Linear Programming Manual

CHENG Gang

(School of Public Health, Peking University)

QIAN Zhenhua

(School of Social Science, University of Science & Technology Beijing)

August 2010

http://www.MaxDEA.cn

Email: MaxDEA@qq.com

Table of Contents

Chapter 1: Export LP equations....................1

1.1 Advantages of LP equations....................1

1.2 How to export LP equations....................1

1.3 Where are the LP equations placed....................2

Chapter 2: Understand LP equations in MaxDEA....................4

2.1 LP equations (formulas) used in MaxDEA....................4

2.1.1 Envelopment model....................4

2.1.1.1 Radial model....................4

2.1.1.2 Non-radial model (SBM)....................8

2.1.2 Multiplier model....................13

2.2 Understand exported LP equations in MaxDEA....................14

2.2.1 File format....................14

2.2.1.1 lp format (*.lp)....................14

2.2.1.2 mps format (*.mps)....................15

2.2.1.3 matrix format (*.txt)....................17

2.2.2 Envelopment model....................20

2.2.2.1 Columns in the exported LP....................20

2.2.2.2 Constraints in the exported LP....................21

2.2.3 Multiplier model....................21

2.2.3.1 Columns in the exported LP....................21

2.2.3.2 Constraints in the exported LP....................21

Appendix 1: Introduction to lpsolve (lp_solve) 5.5 22

A1.1 What is lp_solve 22

A1.2 How to get lp_solve 22

A1.3 Features of lp_solve 22

A1.4 How to use lp_solve 23

Via an IDE 23

Via the API 24

Via input files 24

Appendix 2: Formulas for some DEA models 1

Table 3.1 Radial models 1

Table 3.2 SBM (non-radial) models 2

Table 3.3 Hybrid (mixed) models 3

Table 3.4 Super radial models 4

Table 3.5 Super SBM (non-radial) models* 5

Table 3.6 Super Hybrid (mixed) models* 6

Table 3.7 Bounded Radial models* 7

Table 3.8 Bounded SBM models 8

Table 3.9 Cost, Revenue, Profit and Revenue/Cost Ratio models 9

Table 3.10 Multiplier models 11

Chapter 1: Export LP equations

1.1 Advantages of LP equations

Linear programming equations (LP) equations are the core of DEA models. With MaxDEA, LP equations of DEA models can be exported to matrix format text files (*.txt), lp format text files (*.lp) and mps format text files (*.mps). All the three types of files can be read with text editors, such as Notepad and MS Word. Lp format and mps format files can be opened and solved with the free software lpsolve.

LP equations are useful to help DEA users

1) to have a better understanding of DEA models;
2) to develop DEA models using a linear (mixed) programming software (such as lpsolve and Lingo) or a general language with LP function;
3) to verify the results of DEA models;
4) to develop your own specific DEA models on the basis of the exported LPs.

In addition to LP equations, value of the objective function and values of all variables of the optimal solution are also exported. This function supports all DEA models in MaxDEA.

1.2 How to export LP equations

The function of export LP equations is available in MaxDEA professional edition only.

To export LP equations for your DEA model, you should check the options for this function.

In the tab “Options” , there are three choices for exporting LP equations (Figure 1.3):

1) Export none;
2) Export LP equations for the first DMU only;
3) Export LP equations for all the DMUs.

1.3 Where are the LP equations placed

LP equations are exported to files, and the exported files are placed in the same directory as the MaxDEA program file. The files are named as “LP_” + DMU Name. For example, if the name of a DMU is “A”, the exported file containing its LP equations will be named as “LP_A”. If the “Two stage ” method is selected, the LP equations for the second stage will be exported to a separate file named as “LP_Stage2_” + DMU Name, such as “LP_Stage2_A”.

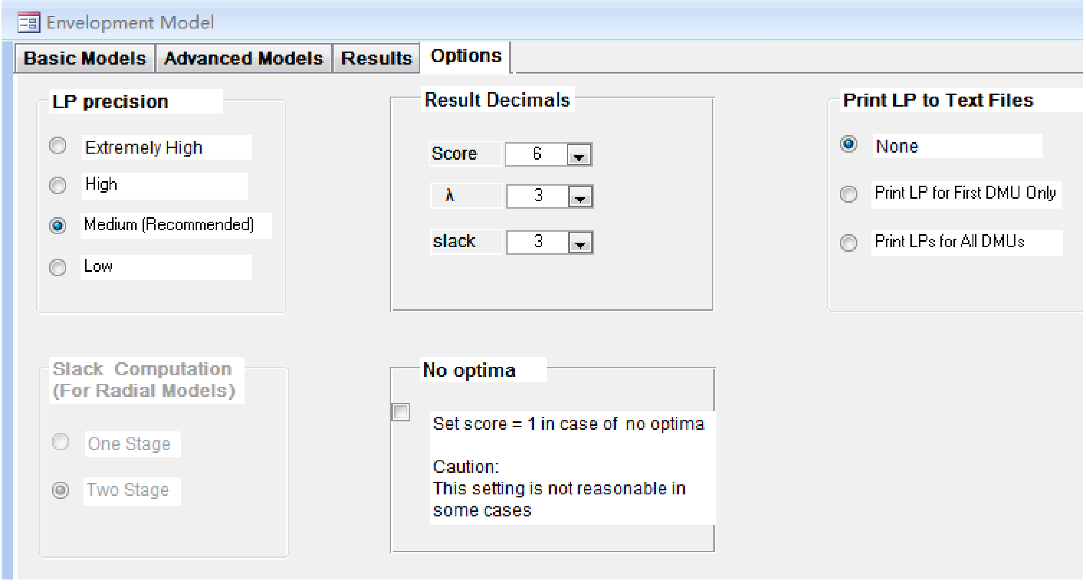


Figure 1.3 Options for Envelopment model

Table 1.3 Name of LP files

DMU Name	File Name for First Stage	File Name for Second Stage
A	LP_A	LP_Stage2_A
B	LP_B	LP_ Stage2_B
C	LP_C	LP_ Stage2_C
D	LP_D	LP_ Stage2_D

For Malmquist model, the LP files are named as follows,

1) The LP files for computing efficiency (t) is named as "LP_" + DMU Name + (t), such as "LP_A(2010)";
2) The LP files for computing efficiency (t-1) is named as "LP_" + DMU Name + (t-1), such as "LP_A(2010-1)";
3) The LP files for computing efficiency (t+1) is named as "LP_" + DMU Name + (t+1), such as "LP_A(2010+1)";

Table 1.3-2 LP files for Mamlquist model

(taking the year 2010 as an example)

DMU Name	LP File Name for Efficiency(2010)	LP File Name for Efficiency(2010-1)	LP File Name for Efficiency(2010+1)
A	LP_A(2010)	LP_A(2010-1)	LP_A(2010+1)
B	LP_B(2010)	LP_B(2010-1)	LP_B(2010+1)
C	LP_C(2010)	LP_C(2010-1)	LP_C(2010+1)
D	LP_D(2010)	LP_D(2010-1)	LP_D(2010+1)

Chapter 2: Understand LP equations in MaxDEA

2.1 LP equations (formulas) used in MaxDEA

2.1.1 Envelopment model

MaxDEA uses generalized equations for DEA models except Cost/ Revenue/Profit/Revenue Cost Ratio models.

Refer to Table 3-6, 3-7, 3-8, 3-9, 3-10, 3-11, 3-12, 3-13, 3-14, 3-15 in “ MaxDEA Manual” for a description of traditional equations.

2.1.1.1 Radial model

The radial model with generalized orientation is expressed as

(n= the number of DMUs; m = the number of inputs; p = the number of outputs)

For DMU_k

$$\min \ \frac{1-w^I\alpha}{1+w^O\beta}$$

$$\text{s.t.} \ \sum_{j=1}^{n}\lambda_j x_{ij}+s_i^- =(1-\alpha)x_{ik} \quad (if \ \ w^I>0)$$

$$\sum_{j=1}^{n}\lambda_j x_{ij}+s_i^- = x_{ik} \quad (if \ \ w^I=0)$$

$$\sum_{j=1}^{n}\lambda_j y_{rj}-s_r^+ =(1+\beta)y_{rk} \quad (if \ \ w^O>0)$$

$$\sum_{j=1}^{n}\lambda_j y_{rj}-s_r^+ = y_{rk} \quad (if \ \ w^O=0)$$

$$\sum_{j=1}^{n}\lambda_j =1 \quad (VRS)$$

$$\lambda_j \ge 0; s_i^- \ge 0; s_r^+ \ge 0$$

j = 1, 2, . . ., n; $i = 1, 2, ..., m;\ r = 1, 2, ..., p$

(1.1.1.1-1)

where w^I and w^O are user-defined non-negative numbers and at least one of them is positive, and the efficiency score is defined as (1-α)/(1+β).

If DMU_k is efficient in the above model, its super-efficiency model is expressed as

$$\min \frac{1 - w^I \alpha}{1 + w^O \beta}$$

$$\text{s.t. } \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} + s_i^- = (1-\alpha) x_{ik} \quad (if\ w^I > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} + s_i^- = (1-\alpha) x_{ik} \quad (if\ w^I = 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} - s_r^+ = (1+\beta) y_{rk} \quad (if\ w^O > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} - s_r^+ = y_{rk} \quad (if\ w^O = 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j = 1 \quad (VRS)$$

$$\alpha \leq 0; \beta \leq 0; \lambda_j \geq 0; s_i^- \geq 0; s_r^+ \geq 0$$

j = 1, 2, . . ., n$(j \neq k)$; $i = 1, 2, ..., m;\ r = 1, 2, ..., p$

(1.1.1.1-2)

where w^I and w^O are user-defined non-negative numbers and at least one of them is positive, and the super-efficiency score is defined as (1-α)/(1+β).

Table 1.1.1.1 reports the relationship between the generalized orientation and the traditional orientations.

Table 1.1.1.1 Special cases of the generalized radial model and their definitions for efficiency score

Case	Orientation	Standard efficiency model			Super-efficiency model		
		w^I	w^O	Score	w^I	w^O	Score
1	**Input-oriented**	**1**	**0**	$1-\alpha^*$	**1**	**0**	$1-\alpha^*$
2	**Output-oriented**	**0**	**1**	$\frac{1}{1+\beta^*}$	**0**	**1**	$\frac{1}{1+\beta^*}$
3	Non-oriented	1	1	$\frac{1-\alpha^*}{1+\beta^*}$	1	1	$\frac{1-\alpha^*}{1+\beta^*}$
4	Input-oriented (modified)	1	ε	$1-\alpha^*$	ε	1	$1-\alpha^*$
6	Non-oriented (input-prioritized)	1	ε	$\frac{1-\alpha^*}{1+\beta^*}$	ε	1	$\frac{1-\alpha^*}{1+\beta^*}$
5	Output-oriented (modified)	ε	1	$\frac{1}{1+\beta^*}$	1	ε	$\frac{1}{1+\beta^*}$
7	Non-oriented (output-prioritized)	ε	1	$\frac{1-\alpha^*}{1+\beta^*}$	1	ε	$\frac{1-\alpha^*}{1+\beta^*}$

ε is the non-Archimedean infinitesimal (in practice, we use 10^-5).

$\theta = 1 - \alpha$; $\varphi = \beta - 1$

The equations of the DEA model with generalized orientation is nonlinear programming, but it can be transformed into the linear programming below using a method similar to the Charnes–Cooper transformation.

By introducing a positive scalar variable t with $t = 1/(1 + w^O\beta)$, the objective function in the standard efficiency model can be transformed into

$$\min \ t - w^I t\alpha$$

and

$$t + \mathrm{w}^{\mathrm{O}} t\beta = 1$$

is added as a constraint.

Now let's define

$$\mathrm{A} = t\alpha, \mathrm{B} = t\beta, \Lambda = t\lambda, S^{-} = ts^{-}, S^{+} = ts^{+}.$$

Then it becomes the following linear program :

$$\min \; t - w^{I}\mathrm{A}$$

$$s.t. \quad t + \mathrm{w}^{\mathrm{O}}\mathrm{B} = 1$$

$$\sum_{j=1}^{n} \Lambda_j x_{ij} + S_i^{-} = (t - \mathrm{A}) x_{ik} \quad (if \;\; w^{I} > 0)$$

$$\sum_{j=1}^{n} \Lambda_j x_{ij} + S_i^{-} = x_{ik} t \quad (if \;\; w^{I} = 0)$$

$$\sum_{j=1}^{n} \Lambda_j y_{rj} - S_r^{+} = (t + \mathrm{B}) y_{rk} \quad (if \;\; w^{O} > 0)$$

$$\sum_{j=1}^{n} \Lambda_j y_{rj} - S_r^{+} = y_{rk} t \quad (if \;\; w^{O} = 0)$$

$$\sum_{j=1}^{n} \Lambda_j = t \quad (VRS)$$

$$t > 0; \Lambda_j \geq 0; S_i^{-} \geq 0; S_r^{+} \geq 0$$

$$\mathrm{j} = 1, 2, \ldots, \mathrm{n}; \;\; i = 1, 2, \ldots, m; \;\; r = 1, 2, \ldots, p$$

(1.1.1.1-3)

Since t is a positive scalar, the transformation is reversible, and the optimal solution is

$\alpha^* = A^*/t^*$, $\beta^* = B^*/t^*$, $\lambda^* = \Lambda^*/t^*$, $s^{-*} = S^{-*}/t^*$, $s^{+*} = S^{+*}/t^*$.

If DMU_k is efficient in the above model, its super-efficiency model can be transformed into the following linear program:

$$\min \; t - w^I \mathrm{A}$$

$$s.t. \;\; t + w^O \mathrm{B} = 1$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \Lambda_j x_{ij} + S_i^- = (1 - \mathrm{A}) x_{ik} \quad (if \;\; w^I > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \Lambda_j x_{ij} + S_i^- = x_{ik} t \quad (if \;\; w^I = 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \Lambda_j y_{rj} - S_r^+ = (1 + \mathrm{B}) y_{rk} \quad (if \;\; w^O > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \Lambda_j y_{rj} - S_r^+ = y_{rk} t \quad (if \;\; w^O = 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \Lambda_j = t \quad (VRS)$$

$$t > 0; \; \mathrm{A} \leq 0, \mathrm{B} \leq 0; \; \Lambda_j \geq 0; \; S_i^- \geq 0; \; S_r^+ \geq 0$$

$$j = 1, 2, .\, j \neq .k, \; n i (= \;\;); \; m \;\; 1r; 2, \ldots,$$

(1.1.1.1-4)

2.1.1.2 Non-radial model (SBM)

The non-radial model with generalized orientation is expressed as

$$\min \quad \rho = \frac{1 - w^{I} (\frac{1}{m} \sum_{i=1}^{m} s_i^- / x_{io})}{1 + w^{O} (\frac{1}{p} \sum_{r=1}^{p} s_r^+ / y_{ro})}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \lambda_j x_{ij} + s_i^- = x_{ik}$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} - s_r^+ = y_{rk}$$

$$\sum_{j=1}^{n} \lambda_j = 1 \quad (VRS)$$

$$\lambda_j \geq 0; s_i^- \geq 0; s_r^+ \geq 0$$

$$j = 1, 2, \ldots, n; \ i = 1, 2, \ldots, m; \ r = 1, 2, \ldots, p$$

(1.1.1.2-1)

If DMU_k is efficient in the above model, its super-efficiency model is expressed as

$$\min \quad \rho = \frac{1 - w^{I} (\frac{1}{m} \sum_{i=1}^{m} s_i^- / x_{io})}{1 + w^{O} (\frac{1}{p} \sum_{r=1}^{p} s_r^+ / y_{ro})}$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} + s_i^- \leq x_{ik}; \quad s_i^- \leq 0 \quad (if \ w^I > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j x_{ij} + s_i^- = x_{ik}; \quad s_i^- \geq 0 \quad (if \ w^I = 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} - s_r^+ \geq y_{rk}; \quad s_r^+ \leq 0 \quad (if \ w^O > 0)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_j y_{rj} - s_r^+ = y_{rk}; \quad s_r^+ \geq 0 \quad (if \ w^O = 0)$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n}\lambda_j = 1 \quad (VRS)$$

$$\lambda_j \geq 0$$

j = 1, 2, . . ., n($j \neq k$); $i = 1, 2, ..., m;\ r = 1, 2, ..., p$

(1.1.1.2-2)

Table 1.1.1.2 Special cases of the generalized non-radial model and their definitions for efficiency score

Case	Orientation	Standard efficiency model			Super-efficiency model		
		w^I	w^O	Score	w^I	w^O	Score
1	**Input-oriented**	**1**	**0**	$1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}$	**1**	**0**	$1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}$
2	**Output-oriented**	**0**	**1**	$\frac{1}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$	**0**	**1**	$\frac{1}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$
3	Non-oriented	1	1	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$	1	1	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$
4	Input-oriented (modified)	1	ε	$1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}$	ε	1	$1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}$
6	Non-oriented (input-prioritized)	1	ε	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$	ε	1	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$
	Output-oriented (modified)	ε	1	$\frac{1}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$	1	ε	$\frac{1}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$
7	Non-oriented (output-prioritized)	ε	1	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$	1	ε	$\frac{1-\frac{1}{m}\sum_{i=1}^{m} s_i^{-*}/x_{io}}{1+\frac{1}{p}\sum_{r=1}^{p} s_r^{+*}/y_{ro}}$

ε is the non-Archimedean infinitesimal (in practice, we use 10^-6).

By introducing a positive scalar variable t with

$$t = \frac{1}{1 + w^{O}(\frac{1}{s}\sum_{r=1}^{s} s_r^{+} / y_{ro})},$$

the objective function in the standard efficiency model can be transformed into

$$\min \quad \rho = t - w^{I}(\tfrac{1}{m}\sum_{i=1}^{m} s_i^{-} t / x_{io})$$

and

$$t + w^{O}(\tfrac{1}{s}\sum_{r=1}^{s} s_r^{+} t / y_{ro}) = 1$$

is added as a constraint.

Now let's define

$$\Lambda = t\lambda, S^{-} = ts^{-}, S^{+} = ts^{+}.$$

Then

it becomes the following linear program:

$$\min \quad \rho = t - w^{I}(\tfrac{1}{m}\sum_{i=1}^{m} S_i^{-} / x_{io})$$

$$\text{s.t.} \quad w^{O}(\tfrac{1}{s}\sum_{r=1}^{s} S_r^{+} / y_{ro}) + t = 1$$

$$\sum_{j=1}^{n} \Lambda_j x_{ij} + S_i^{-} = x_{ik} t$$

$$\sum_{j=1}^{n} \Lambda_j y_{rj} - S_r^{+} = y_{rk} t$$

$$\sum_{j=1}^{n} \Lambda_j = t \quad (VRS)$$

$$\Lambda_j \geq 0; S_i^- \geq 0; S_r^+ \geq 0$$

$$\text{j = 1, 2, . . ., n; } i = 1, 2, ..., m;\ r = 1, 2, ..., p$$

(1.1.1.2-3)

Since t is a positive scalar, the transformation is reversible, and the optimal solution is

$\alpha^* = A^*/t^*$, $\beta^* = B^*/t^*$, $\lambda^* = \Lambda^*/t^*$, $s^{-*} = S^{-*}/t^*$, $s^{+*} = S^{+*}/t^*$.

If DMU_k is efficient in the above model, its super-efficiency model can be transformed into the following linear program:

$$\min\ \ \rho = t - w^I \left(\tfrac{1}{m}\sum_{i=1}^{m} S_i^- / x_{io}\right)$$

$$\text{s.t.}\ \ w^O \left(\tfrac{1}{s}\sum_{r=1}^{s} S_r^+ / y_{ro}\right) + t = 1$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n} \Lambda_j x_{ij} + S_i^- \leq x_{ik}t;\quad S_i^- \leq 0 \quad (if\ \ w^I > 0)$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n} \Lambda_j x_{ij} + S_i^- = x_{ik}t\,;\ S_i^- \geq 0\ \ (if\ \ w^I = 0)$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n} \Lambda_j y_{rj} - S_r^+ \geq y_{rk}t,\quad r = 1, 2, ..., p;\ S_r^+ \leq 0,\ \ r = 1, 2, ..., p\ (if\ \ w^O > 0)$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n} \Lambda_j y_{rj} - S_r^+ = y_{rk}t,\quad r = 1, 2, ..., p;\ S_r^+ \geq 0,\ \ r = 1, 2, ..., p\ (if\ \ w^O = 0)$$

$$\sum_{\substack{j=1\\ j\neq k}}^{n} \Lambda_j = t \quad (VRS)$$

$$\Lambda_j \geq 0$$

$$\text{j = 1, 2, . . ., n}(j \neq k);\ i = 1, 2, ..., m;\ r = 1, 2, ..., p$$

(1.1.1.2-4)

2.1.2 Multiplier model

Table 2.1.2 Multiplier models

Orientation	**Input-oriented**	**Output-oriented**
	$\max \theta = \mu' y_0 + \mu_0$	$\max \phi = \nu' x_0 + \nu_0$
	$st \;\; \nu' x_0 = 1$	$st \;\; \mu' y_0 = 1$
	$-\nu'\mathrm{X} + (\mu'\mathrm{Y} + \mu_0) \le 0$	$-(\nu'\mathrm{X} + \nu_0) + \mu'\mathrm{Y} \le 0$
	$\mu, \nu \ge 0$ (or $\mu, \nu \ge \varepsilon$)*	$\mu, \nu \ge 0$ (or $\mu, \nu \ge \varepsilon$)*
CRS	$\mu_0 = 0$	$\nu_0 = 0$
VRS	μ_0 *free*	ν_0 *free*
NIRS	$\mu_0 \le 0$	$\nu_0 \ge 0$
NDRS	$\mu_0 \ge 0$	$\nu_0 \le 0$
GRS**	$L \le \mu_0 \le U$	$L \le \nu_0 \le U$

*: ε is a user-defined parameter, ε≥0. If ε is not properly specified, such as a big number, the LP may be infeasible.

**: L and U are user-defined parameters, L≤U.

2.2 Understand exported LP equations in MaxDEA

2.2.1 File format

2.2.1.1 lp format (*.lp)

To understand the exported LP equations, the lp format file (*.lp) should be the best choice. It is very readable and its syntax is very similar to the mathematical formulation. The lp format file can open with any text editor programs (such as Word, Notepad). Note that open the lp file, you should click the menu "Open", and select " All files" in the file type list in the "open file dialogue box ".

The lp format is the native lpsolve format to provide LP models via an ASCII file to the solver. It can be directly read by the linear (mixed) programming software lpsolve. It is very readable and its syntax is very similar to the Mathematical formulation. An example model in lp-format:

Example 1:

```
/* model.lp */

max: 143 x + 60 y;

120 x + 210 y <= 15000;
110 x + 30 y <= 4000;
x + y <= 75;
```

Example 2:

```
/* DMU_C */

/* Objective function */
min: +t -ALPHA;

/* Constraints */
Constraint_t: +t = 1;
Constraint_Input1: -0.321 t +0.321 ALPHA +0.838 LAMBDA_A +1.233 LAMBDA_B +0.321
LAMBDA_C +1.483 LAMBDA_D +SLACK_Input1 = 0;
Constraint_Output1: -0.5545 t +0.608 LAMBDA_A +0.359 LAMBDA_B +0.5545 LAMBDA_C
+0.5305 LAMBDA_D -SLACK_Output1 = 0;
```

2.2.1.2 mps format (*.mps)

If you want to solve the model in a linear program such as [lpsolve](), Lingo, the mps format should be the best choice.

The MPS format is supported by most lp solvers and thus very universal. The model is provided to the solver via an ASCII file. But this format is very difficult to read by humans. An example model in MPS format:

Example 1:

```
* model.mps
NAME
ROWS
 N   R0
 L   R1
 L   R2
 L   R3
COLUMNS
    x         R0        143.00000000    R1        120.00000000
    x         R2        110.00000000    R3        1.0000000000
    y         R0        60.00000000     R1        210.00000000
    y         R2        30.000000000    R3        1.0000000000
RHS
    RHS       R1        15000.000000    R2        4000.0000000
    RHS       R3        75.000000000
ENDATA
```

Example 2:

```
*<meta creator='lp_solve v5.5'>

*<meta rows=5>

*<meta columns=10>

*<meta equalities=3>

*<meta integers=0>

*<meta origsense='MIN'>

*

NAME          DMU_C
```

```
ROWS
 N  R0
 E  Constraint_t
 E  Constraint_Input1
 E  Constraint_Output1
 L  Constraint_ALPHA_SuperEfficiency
 L  Constraint_BETA_SuperEfficiency
COLUMNS
    t          R0            1.0000000000    Constraint_t   1.0000000000
    t          Constraint_Input1   -0.321000000    Constraint_Output1   -0.554500000
    ALPHA      R0            -1.000000000    Constraint_Input1   0.3210000000
    ALPHA      Constraint_ALPHA_SuperEfficiency   1.0000000000
    BETA       Constraint_BETA_SuperEfficiency   1.0000000000
    LAMBDA_A   Constraint_Input1   0.8380000000    Constraint_Output1   0.6080000000
    LAMBDA_B   Constraint_Input1   1.2330000000    Constraint_Output1   0.3590000000
    LAMBDA_D   Constraint_Input1   1.4830000000    Constraint_Output1   0.5305000000
    SLACK_Input1   Constraint_Input1   1.0000000000
    SLACK_Output1   Constraint_Output1   -1.000000000
RHS
    RHS        Constraint_t   1.0000000000
BOUNDS
 FR BND        ALPHA
 FR BND        BETA
 FR BND        GAMA
ENDATA
```

There are two kinds of mps formats: one is fixed mps format, and the other is free mps format. The free format is very similar to the fixed MPS format, but it is less restrictive e.g. it allows longer names. The free mps format that MaxDEA exports is the same as the fixed format, except that the free format permits names longer than 8 characters.

2.2.1.3 matrix format (*.txt)

In addition to matrix format LP equations, the exported text files (*.txt) provide more information.

To explain the exported text files, a simple dataset with one input, one output, and 4 DMUs will be used for demonstration:

DMU	Input1	Output1
A	0.838	0.608
B	1.233	0.359
C	0.321	0.5545
D	1.483	0.5305

Contents of the exported text files

The exported text file has 4 parts:

(1) Introduction

A sentence indicating Which DMU is evaluated, and if it is a super-efficiency model, there will be another sentence indicating that the LP is for a super-efficiency model.

Example:

Linear program for DMU: C

The linear program here is for calculating super-efficiency for DMU: C

Please see the 'MaxDEA Manual' for a description of the linear program!

(2) LP matrix;

Example 1:

Model name:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10		
Minimize	1	-1	0	0	0	0	0	0	0	0		
R1	1	0	0	0	0	0	0	0	0	0	=	1
R2	-0.321	0.321	0	0	0.838	1.233	0	1.483	1	0	=	0
R3	-0.5545	0	0	0	0.608	0.359	0	0.5305	0	-1	=	0

R4	0	1	0	0	0	0	0	0	0	0	<=	0
R5	0	0	1	0	0	0	0	0	0	0	<=	0
Type	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real		
upbo	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf		
lowbo	0	-Inf	-Inf	-Inf	0	0	0	0	0	0		

Example 2:

Model name: DMU_C

	t	ALPHA	BETA	GAMA	LAMBDA_A	LAMBDA_B	LAMBDA_C	LAMBDA_D	SLACK_Input1	SLACK_Output1		
Minimize	1	-1	0	0	0	0	0	0	0	0		
Constraint_t	1	0	0	0	0	0	0	0	0	0	=	1
Constraint_Input1	-0.321	0.321	0	0	0.838	1.233	0.321	1.483	1	0	=	0
Constraint_Output1	-0.5545	0	0	0	0.608	0.359	0.5545	0.5305	0	-1	=	0
Type	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real		
upbo	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf		
lowbo	0	0	0	0	0	0	0	0	0	0		

In the LP matrix, each column is an variable named according to its meaning (here, denoted as C1, C2, ….) The First row is the objective function (minimize or maximize), the following rows are constraints named according to their meanings (here, denoted as R1, R2, …) and the last three rows indicate the type (real or integer) and the upper and lower bounds of each variable.

The above equations (example 1) can be transformed as

Min: C1 - C2

s.t

1*C1										=	1
-0.321*C1+	0.321*C2+	0*C3+	0*C4+	0.838*C5+	1.233*C6+		1.483*C8+	1*C9+		=	0
-0.5545*C1+				0.608*C5+	0.359*C6+		0.5305*C8+		-1*C10	=	0
	1*C2									<=	0
		1*C3								<=	0
1*C1										>=	0
				1*C5						>=	0
					1*C6					>=	0
										>=	0
							1*C8+			>=	0
								1*C9+		>=	0
									1*C10	>=	0

You can export the LP matrix to Excel sheet as follows,

Copy the equations to Excel 2003

- Copy the text to excel sheet;
- Select (highlight) the text or the column;
- Click the menu "Data - Text to Columns...";
- Choose the file type that best describes your data: Select "Delimited", Next;
- Check the box "Space" and the box "Treat consecutive delimiters as one", Finish.

Copy the equations to Excel 2007

- Copy the text to excel sheet;
- Select (highlight) the text or the column;
- Click the button “Text to Columns" on the tab "Data”;
- Choose the file type that best describes your data: Select "Delimited", Next;
- Check the box "Space" and the box "Treat consecutive delimiters as one", Finish.

There is an easier way to get the LP matrix in Excel:

- Open the lp or mps format file with LpSolve (IDE);
- Solve the LP;
- Export the LP matrix to rtf file(File – Export Matrix- To RTF;
- Open the RTF file, and copy the table to Excel.

(3) The objective value of the LP;

This part reports the value of the objective function. The objective value is the efficiency score of the evaluated DMU in DEA models with traditional orientations (input-, output- and non- orientation) except output-oriented multiplier model and Cost/Revenue/Profit/ Cost Revenue Ratios model.

Example

The objective value of the lp:

Value of objective function: 2.38087699

(4) The solution of the LP.

This parts reports actual values of the variables.

The solution of the lp:

Actual values of the variables:

t	1
ALPHA	-1.38088
BETA	0
GAMA	0
LAMBDA_A	0.912007
LAMBDA_B	0
LAMBDA_C	0
LAMBDA_D	0
SLACK_Input1	0
SLACK_Output1	0

2.2.2 Envelopment model

2.2.2.1 Columns in the exported LP

1) The first column is for the positive scalar variable *t*;
2) The second column is for A with the relationship $\theta = 1 - A/t$;
3) The third column is for B with the relationship $\varphi = 1 + B/t$;
4) The following n columns are for Λ with the relationship $\lambda = \Lambda/t$;
5) The rest are for S^- and S^+ with the relationship $s^- = S^{-/}t$ and $s^+ = S^{+/}t$.

Table 1.2.1.1 The relationship between the column in the exported LP and the variables in the DEA model

Column in the exported LP	Variable in the original equations	Variable in the DEA model in the literature
t		
ALPHA (A)	$\alpha = A/t$	$\theta = 1 - A/t$
BETA (B)	$\beta = B/t$	$\varphi = 1 + B/t$
GAMA (Γ)	$\gamma = \Gamma/t$	$\gamma = \Gamma/t$
LAMBDA (Λ)	$\lambda = \Lambda/t$	$\lambda = \Lambda/t$

SLACK_Input (S^-)	$s^- = S^{-/}t$	$s^- = S^{-/}t$
SLACK_Output (S^+)	$s^+ = S^{+/}t$	$s^+ = S^{+/}t$

γ is used in undesired radial models (directional distance function)

2.2.2.2 Constraints in the exported LP

The constraints in the exported LP include

1) The first constraint for the positive scalar variable *t*;
2) Constraints for inputs;
3) Constraints for outputs;
4) Constraints for Λ in non-CRS models;
5) Constraints for A in radial super-efficiency models;
6) Constraints for B in radial super-efficiency models;
7) Constraints for S- in non-radial super-efficiency models;
8) Constraints for S+ in non-radial super-efficiency models;

2.2.3 Multiplier model

2.2.3.1 Columns in the exported LP

1) The first m column are for ν;
2) The following p columns are for μ;
3) The last column is for the free variable ν_0 or μ_0.

2.2.3.2 Constraints in the exported LP

The constraints in the exported LP include

1) The first constraint for

   $\nu' x_0 = 1$ (input-oriented) or

   $\mu' y_0 = 1$ (output-oriented);

2) N (n-1 in super-efficiency models) constraints for

   $(\mu' \mathrm{Y} + \mu_0) - \nu' \mathrm{X} \le 0$ (input-oriented) or

   $\mu' \mathrm{Y} - (\nu' \mathrm{X} + \nu_0) \le 0$ (output-oriented);

3) Constraints for restrictions in restricted multiplier models;

Appendix 1: Introduction to lpsolve (lp_solve) 5.5

The following contents are copied from the help file of lpsolve 5.5

A1.1 What is lp_solve

What is lp_solve and what is it not? The simple answer is, lp_solve is a Mixed Integer Linear Programming (MILP) solver.

lp_solve is a **free** (see LGPL for the GNU lesser general public license) linear (integer) programming solver based on the revised simplex method and the Branch-and-bound method for the integers.

A1.2 How to get lp_solve

All needed files can be downloaded from sourceforge on the following location: http://sourceforge.net/projects/lpsolve/

It contains full source, examples and manuals.

A1.3 Features of lp_solve

Here is a list of some key features of lp_solve:

- Mixed Integer Linear Programming (MILP) solver
- Basically no limit on model size
- It is free and with sources
- Supports Integer variables, Semi-continuous variables and Special Ordered Sets
- Can read model from MPS, LP or user written format
- Models can be build in-memory without the use of files
- Has a powerful API interface
- Easy callable from other programming languages
- Advanced pricing using Devex and Steepest Edge for both primal and dual simplexes
- Provides different scaling methods to make the model more numerical stable
- Has presolve capabilities to tighten constraints/make the model smaller and faster to solve
- Has a base crashing routine to determine a starting point
- Allows restart after making changes to the model. Solve continues from the last found solution

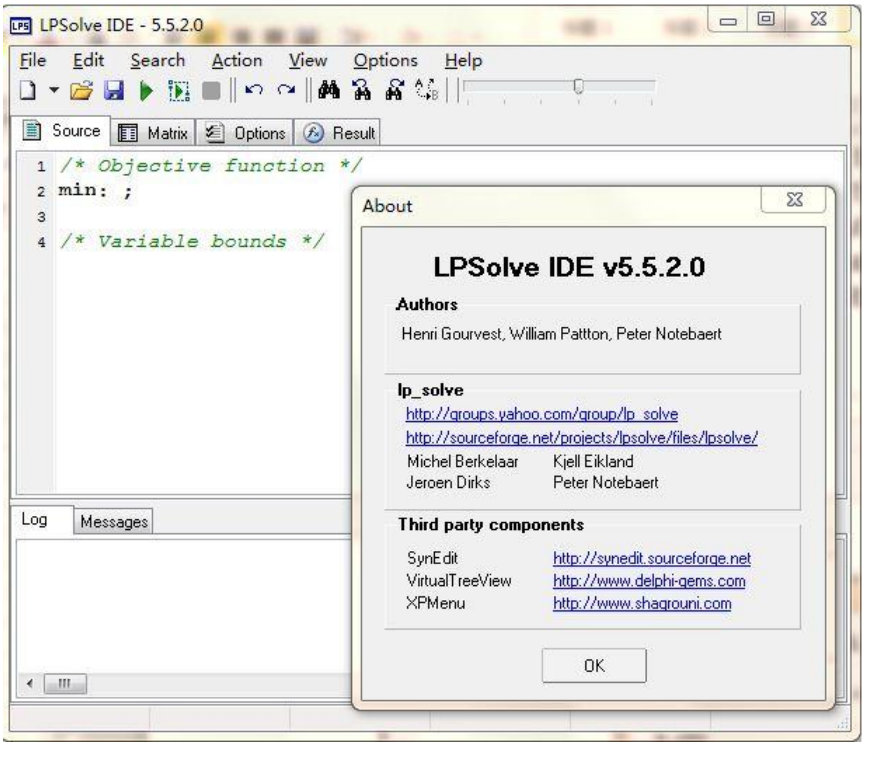
- Possible to select desired combinations of primal and dual phases 1 and 2
- Possible to set several solver parameters like tolerances
- Alternative (and faster) inverse/re-factorisation libraries are provided for. See Basis Factorization Packages
- Alternative model readers and writers possible via the XLI implementation. See External Language Interfaces
- Has the possibility to convert one model format to another format
- Provides post-optimal sensitivity analysis. See Sensitivity
- ...

A1.4 How to use lp_solve

Basically, lp_solve is a library, a set of routines, called the API that can be called from almost any programming language to solve MILP problems. There are several ways to pass the data to the library:

- Via an IDE
- Via the API
- Via input files

Via an IDE



- Thanks to Henri Gourvest, there is now also an IDE program called LPSolve IDE that uses the API to provide a Windows application to solve models. See LPSolve IDE for its usage. With this program you don't have to know anything of API or computer programming languages. You can just provide your model to the program and it will solve the model and give you the result.

Via the API

The API is a set of routines that can be called from a programming language to build the model in memory, solve it and return the results. There are many API routines to perform many possible tasks and set several options. See lp_solve API reference for an overview.

As already stated, lp_solve can be called from many programming language. Among them are C, C++, Pascal, Delphi, Java, VB, C#, VB.NET, Excel. But let this list not be a limitation. Any programming language capable of calling external libraries (DLLs under Windows, Shared libraries (.so) under Unix/Linux) can call lp_solve.

Via input files

Standard, lp_solve supports several input files types. The common known MPS format (see mps-format) is supported by most solvers, but it is not very readable for humans. Another format is the lp format (see lp-format) that is more readable. lp_solve has the unique ability to use user-written routines to input the model (see External Language Interface). See read_mps, read_freemps, read_MPS, read_freeMPS and read_lp, read_LP for the API calls to read the model from file.

There is also a driver program called lp_solve that uses the API to provide a command line application to solve models. See lp_solve for its usage. With this program you don't have to know anything of API or computer programming languages. You can just provide your model via file to the program and it will solve the model and give you the result.

Appendix 2: Formulas for some DEA models

Table 3.1 Radial models

Orientation	**Input-oriented**	**Output-oriented**	**Nonoriented**
CRS	$\min\ \theta$ $st\ \ \theta x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\max \phi$ $st\ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - \phi y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\min\ \theta/\phi$ $st\ \ \theta x_0 - X\lambda - s^- = 0$ $Y\lambda - \phi y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \leq 1$	$e\lambda \leq 1$	$e\lambda \leq 1$
NDRS	$e\lambda \geq 1$	$e\lambda \geq 1$	$e\lambda \geq 1$
GRS*	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$

*: L and U are user-defined parameters, L≥0, U≥0, L≤U.

Table 3.2 SBM (non-radial) models

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min \ \rho = 1 - \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}$ $st \ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\min \ \rho = \dfrac{1}{1 + \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st \ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\min \ \rho = \dfrac{1 - \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}}{1 + \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st \ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \leq 1$	$e\lambda \leq 1$	$e\lambda \leq 1$
NDRS	$e\lambda \geq 1$	$e\lambda \geq 1$	$e\lambda \geq 1$
GRS	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$

Table 3.3 Hybrid (mixed) models

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min \rho = 1 - \frac{m_1}{m}(1-\theta) - \frac{1}{m}\sum_{i=1}^{m_2} s_i^{NR-} / x_{io}^{NR}$ $st\ \ \theta x_0^R - X^R\lambda - s^{R-} = 0$ $x_0^{NR} - X^{NR}\lambda - s^{NR-} = 0$ $Y\lambda - y_0 - s^+ = 0$ $\theta \le 1, \lambda, s^-, s^+ \ge 0$	$\min \rho = \frac{1}{1 + \frac{s_1}{s}(\phi - 1) + \frac{1}{s}\sum_{r=1}^{s_2} s_r^{NR+} / y_{ro}^{NR}}$ $st\ \ x_0 - X\lambda - s^- = 0$ $Y^R\lambda - \phi y_0^R - s^{R+} = 0$ $Y^{NR}\lambda - y_0^{NR} - s^{NR+} = 0$ $\phi \ge 1,\ \lambda, s^-, s^+ \ge 0$	$\min \rho = \frac{1 - \frac{m_1}{m}(1-\theta) - \frac{1}{m}\sum_{i=1}^{m_2} s_i^{NR-} / x_{io}^{NR}}{1 + \frac{s_1}{s}(\phi - 1) + \frac{1}{s}\sum_{r=1}^{s_2} s_r^{NR+} / y_{ro}^{NR}}$ $st\ \ \theta x_0^R - X^R\lambda - s^{R-} = 0$ $x_0^{NR} - X^{NR}\lambda - s^{NR-} = 0$ $Y^R\lambda - \phi y_0^R - s^{R+} = 0$ $Y^{NR}\lambda - y_0^{NR} - s^{NR+} = 0$ $\theta \le 1, \phi \ge 1,\ \lambda, s^-, s^+ \ge 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \le 1$	$e\lambda \le 1$	$e\lambda \le 1$
NDRS	$e\lambda \ge 1$	$e\lambda \ge 1$	$e\lambda \ge 1$
GRS	$L \le e\lambda \le U$	$L \le e\lambda \le U$	$L \le e\lambda \le U$

Table 3.4 Super radial models

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min\ \theta$ $st\ \ \theta x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j - s^- = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\max \phi$ $st\ \ x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j - s^- = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - \phi y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\min\ \theta/\phi$ $st\ \ \theta x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j - s^- = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - \phi y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \leq 1$	$e\lambda \leq 1$	$e\lambda \leq 1$
NDRS	$e\lambda \geq 1$	$e\lambda \geq 1$	$e\lambda \geq 1$
GRS	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$

Table 3.5 Super SBM (non-radial) models*

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min\ \rho_{super} = 1 + \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}$ $st\ \ x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j + s^- \geq 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - y_0 - s^+ = 0$ $\lambda, s^-, s^+ \geq 0$	$\min \rho_{super} = \dfrac{1}{1 - \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st\ \ x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j - s^- = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - y_0 + s^+ \geq 0$ $\lambda, s^-, s^+ \geq 0$	$\min \rho_{super} = \dfrac{1 + \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}}{1 - \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st\ \ x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j + s^- \geq 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - y_0 + s^+ \geq 0$ $\lambda, s^-, s^+ \geq 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \leq 1$	$e\lambda \leq 1$	$e\lambda \leq 1$
NDRS	$e\lambda \geq 1$	$e\lambda \geq 1$	$e\lambda \geq 1$
GRS	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$

*: The LP equations are used to calculate the "super efficiency" only.

Table 3.6 Super Hybrid (mixed) models*

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min \rho_{\text{super}} = 1 + \frac{m_1}{m}(\theta - 1) + \frac{1}{m}\sum_{i=1}^{m_2} s_i^{NR-} / x_{io}^{NR}$ $st\ \ \theta x_0^R - \sum_{j=1,\neq 0}^{n} \lambda_j x_j^R - s^{R-} = 0$ $x_0^{NR} - \sum_{j=1,\neq 0}^{n} \lambda_j x_j^{NR} + s^{NR-} \geq 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j - y_0 - s^{+} = 0$ $\theta \geq 1, \lambda, s^{-}, s^{+} \geq 0$	$\min \rho_{\text{super}} = \dfrac{1}{1 - \frac{s_1}{s}(1-\phi) - \frac{1}{s}\sum_{r=1}^{s_2} s_r^{NR+} / y_{ro}^{NR}}$ $st\ \ x_0 - \sum_{j=1,\neq 0}^{n} \lambda_j x_j - s^{-} = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j^R - \phi y_0^R - s^{R+} = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j^{NR} - y_0^{NR} + s^{NR+} \geq 0$ $\phi \leq 1,\ \ \lambda, s^{-}, s^{+} \geq 0$	$\min \rho_{\text{super}} = \dfrac{1 + \frac{m_1}{m}(\theta - 1) + \frac{1}{m}\sum_{i=1}^{m_2} s_i^{NR-} / x_{io}^{NR}}{1 - \frac{s_1}{s}(1-\phi) - \frac{1}{s}\sum_{r=1}^{s_2} s_r^{NR+} / y_{ro}^{NR}}$ $st\ \ \theta x_0^R - \sum_{j=1,\neq 0}^{n} \lambda_j x_j^R - s^{R-} = 0$ $x_0^{NR} - \sum_{j=1,\neq 0}^{n} \lambda_j x_j^{NR} + s^{NR-} \geq 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j^R - \phi y_0^R - s^{R+} = 0$ $\sum_{j=1,\neq 0}^{n} \lambda_j y_j^{NR} - y_0^{NR} + s^{NR+} \geq 0$ $\theta \geq 1, \phi \leq 1,\ \ \lambda, s^{-}, s^{+} \geq 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \leq 1$	$e\lambda \leq 1$	$e\lambda \leq 1$
NDRS	$e\lambda \geq 1$	$e\lambda \geq 1$	$e\lambda \geq 1$
GRS	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$	$L \leq e\lambda \leq U$

*: The LP equations are used to calculate the “super efficiency” only

Table 3.7 Bounded Radial models*

Orientation	**Input-oriented**	**Output-oriented**	**Nonoriented**
CRS	$\min\ \theta$ $st\ \theta x_0^F - X^F\lambda - s^{F-} = 0$ $x_0^B - X^B\lambda - s^{B-} = 0$ $Y\lambda - y_0 - s^+ = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$	$\max\ \phi$ $st\ x_0 - X\lambda - s^- = 0$ $Y^F\lambda - \phi y_0^F - s^{F+} = 0$ $Y^B\lambda - y_0^B - s^{B+} = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$	$\min\ \theta/\phi$ $st\ \theta x_0^F - X^F\lambda - s^{F-} = 0$ $x_0^B - X^B\lambda - s^{B-} = 0$ $Y^F\lambda - \phi y_0^F - s^{F+} = 0$ $Y^B\lambda - y_0^B - s^{B+} = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \le 1$	$e\lambda \le 1$	$e\lambda \le 1$
NDRS	$e\lambda \ge 1$	$e\lambda \ge 1$	$e\lambda \ge 1$
GRS	$L \le e\lambda \le U$	$L \le e\lambda \le U$	$L \le e\lambda \le U$

*The superscript “F” indicates full discretion and “B” indicates bounded.
LB indicates lower bound, and UB indicates upper bound.

Table 3.8 Bounded SBM models

Orientation	Input-oriented	Output-oriented	Nonoriented
CRS	$\min\ \rho = 1 - \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}$ $st\ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$	$\min\ \rho = \frac{1}{1 + \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st\ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$	$\min\ \rho = \frac{1 - \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}}{1 + \frac{1}{s}\sum_{r=1}^{s} s_r^+ / y_{ro}}$ $st\ \ x_0 - X\lambda - s^- = 0$ $Y\lambda - y_0 - s^+ = 0$ $LB \le X^B\lambda$ $Y^B\lambda \le UB$ $\lambda, s^-, s^+ \ge 0$
VRS	$e\lambda = 1$	$e\lambda = 1$	$e\lambda = 1$
NIRS	$e\lambda \le 1$	$e\lambda \le 1$	$e\lambda \le 1$
NDRS	$e\lambda \ge 1$	$e\lambda \ge 1$	$e\lambda \ge 1$
GRS	$L \le e\lambda \le U$	$L \le e\lambda \le U$	$L \le e\lambda \le U$

Table 3.9 Cost, Revenue, Profit and Revenue/Cost Ratio models

	Type I	Type II
Cost	$\text{Cost Efficiency} = \dfrac{cx^*}{cx_0}$ $cx^* = \min\ cx$ $st\ \ x \geq X\lambda$ $y_0 \leq Y\lambda$ $\lambda \geq 0$	$\text{Cost Efficiency} = \dfrac{e\bar{x}^*}{e\bar{x}_0}$ $e\bar{x}^* = \min\ e\bar{x}$ $st\ \ \bar{x} \geq \bar{X}\lambda$ $y_0 \leq Y\lambda$ $\lambda \geq 0$
Revenue	$\text{Revenue Efficiency} = \dfrac{ry_0}{ry^*}$ $ry^* = \max\ ry$ $st\ \ x_0 \geq X\lambda$ $y \leq Y\lambda$ $\lambda \geq 0$	$\text{Revenue Efficiency} = \dfrac{e\bar{y}_0}{e\bar{y}^*}$ $e\bar{y}^* = \max\ e\bar{y}$ $st\ \ x_0 \geq X\lambda$ $\bar{y} \leq \bar{Y}\lambda$ $\lambda \geq 0$

Profit	$\text{Revenue Efficiency} = \dfrac{ry_0 - cx_0}{ry^* - cx^*}$ $ry^* - cx^* = \max \ ry - cx$ $st \ \ x = X\lambda \le x_0$ $y = Y\lambda \ge y_0$ $\lambda \ge 0$	$\text{Revenue Efficiency} = \dfrac{e\bar{y}_0 - e\bar{x}_0}{e\bar{y}^* - e\bar{x}^*}$ $e\bar{y}^* - e\bar{x}^* = \max \ e\bar{y} - e\bar{x}$ $st \ \ \bar{x} = \bar{X}\lambda \le \bar{x}_0$ $\bar{y} = \bar{Y}\lambda \ge \bar{y}_0$ $\lambda \ge 0$
Revenue/Cost Ratio	$\text{Revenue Ratio Efficiency} = \dfrac{ry_0 / cx_0}{ry^* / cx^*}$ $\dfrac{ry^*}{cx^*} = \max \ \dfrac{ry}{cx}$ $st \ \ x = X\lambda \le x_0$ $y = Y\lambda \ge y_0$ $\lambda \ge 0$	$\text{Revenue Ratio Efficiency} = \dfrac{e\bar{y}_0 / e\bar{x}_0}{e\bar{y}^* / e\bar{x}^*}$ $\dfrac{e\bar{y}^*}{e\bar{x}^*} = \max \ \dfrac{e\bar{y}}{e\bar{x}}$ $st \ \ \bar{x} = \bar{X}\lambda \le \bar{x}_0$ $\bar{y} = \bar{Y}\lambda \ge \bar{y}_0$ $\lambda \ge 0$

$\bar{x}_{i,j}$ is the element of matrix $\bar{X}$: $\bar{x}_{i,j} = c_{i,j} \times x_{i,j}$, $\quad c_{i,j}$ *is the price of* $x_{i,j}$

$\bar{y}_{i,j}$ is the element of matrix $\bar{Y}$: $\bar{y}_{i,j} = r_{i,j} \times y_{i,j}$, $\quad r_{i,j}$ *is the price of* $y_{i,j}$

Table 3.10 Multiplier models

Orientation	Input-oriented	Output-oriented
	$\max \theta = \mu' y_0 + \mu_0$ $st \;\; \nu' x_0 = 1$ $(\mu' \mathrm{Y} + \mu_0) - \nu' \mathrm{X} \le 0$ $\mu, \nu \ge 0$ (or $\mu, \nu \ge \varepsilon$)*	$\min \phi = \nu' x_0 + \nu_0$ $st \;\; \mu' y_0 = 1$ $\mu' \mathrm{Y} - (\nu' \mathrm{X} + \nu_0) \le 0$ $\mu, \nu \ge 0$ (or $\mu, \nu \ge \varepsilon$)*
CRS	$\mu_0 = 0$	$\nu_0 = 0$
VRS	μ_0 *free*	ν_0 *free*
NIRS	$\mu_0 \le 0$	$\nu_0 \ge 0$
NDRS	$\mu_0 \ge 0$	$\nu_0 \le 0$
GRS**	$L \le \mu_0 \le U$	$L \le \nu_0 \le U$

*: ε is a user-defined parameter, ε≥0. If ε is not properly specified, such as a big number, the LP may be infeasible.

**: L and U are user-defined parameters, L≤U.

This manual will be updated to include more details about the advanced DEA models.