

Misurazione e valutazione delle *performance*

Lezione 5

DEA: MODELLI CRS E VRS CON SLACK

Marialisa Mazzocchitti

Università degli Studi «G. d'Annunzio» di Chieti-Pescara

Agenda

1. Modelli DEA CRS con slack
2. Modelli DEA VRS con slack

Modelli DEA CRS con *slack*

Dov'è lo *slack*? (1/3)

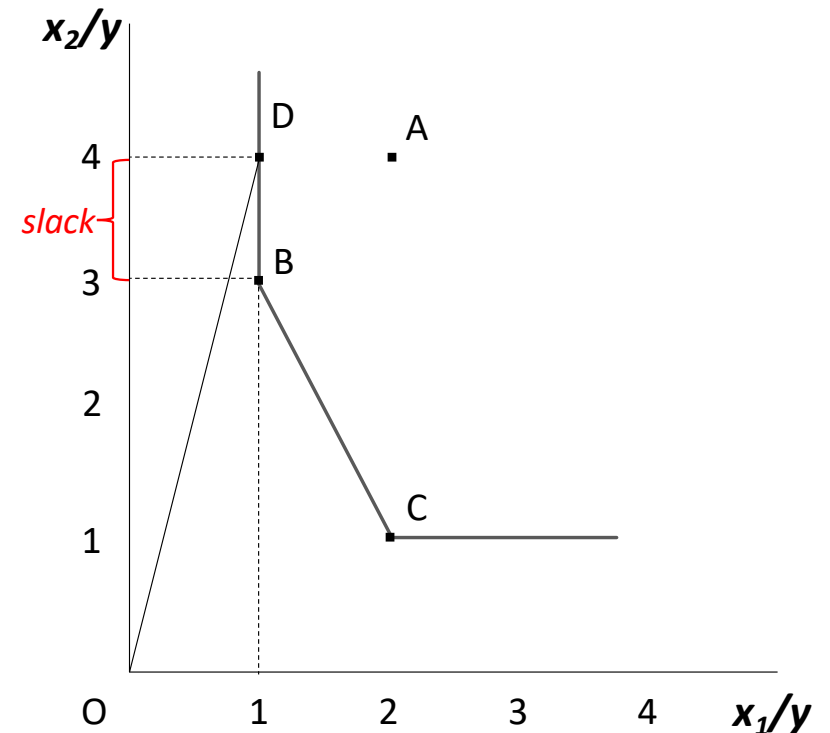
- Gli esempi presentati nel corso del precedente incontro sono stati definiti «semplici» sia perché si prendevano in considerazione soltanto 3 DMU che realizzano un processo produttivo caratterizzato da due input e un output o da un input e due output, sia perché anche nella presentazione del generico modello DEA-CRS (estensione a K DMUs, N outputs, M inputs) non si è tenuto conto di possibili *slack*
- In altre parole, oltre alle λ abbiamo calcolato il valore di θ nel primo esempio e ϕ nel secondo che corrispondono alla misura di input-efficienza di Debreu-Farrell e alla misura di output-efficienza di Debreu-Farrell rispettivamente
- Ma uno dei principali limiti della misura di Debreu-Farrell è che un'unità produttiva può avere un punteggio di efficienza pari a 1 ed essere ancora inefficiente nel senso di Koopmans
 - Questo accade quando una unità si trova in una condizione tale per cui una contrazione (espansione) radiale degli input (output) non è possibile ma soltanto uno o alcuni degli input (output) tra tutti potrebbero ulteriormente essere ridotti (espansi) senza influire sulla necessità di altri input (o della produzione di altri output)
 - Tale eccesso di uno o alcuni input e, parimenti, mancanza/deficit di uno o alcuni output viene definito *slack*

Dov'è lo *slack*? (2/3)

Si prenda in considerazione l'unità D : la sua misura di input-efficienza assume valore pari a 1 ($\overline{OD}/\overline{OD}$), infatti questa non può ridurre in senso radiale i livelli di input impiegati nel processo produttivo e continuare a produrre l'output y , ma a parità di output ha impiegato lo stesso livello di input x_1 impiegato dall'unità B e un eccesso di input x_2 (*slack*) rispetto all'unità produttiva B

Pertanto l'unità D non è efficiente in senso di Koopmans (se riducesse il livello del solo input x_2 fino al livello impiegato dall'unità B riuscirebbe comunque a produrre l'output y)

In generale, in corrispondenza di tutte le unità che si collocano sul tratto verticale dell'isoquanto e quelle che si collocano sul tratto orizzontale dello stesso, la misura di input-efficienza di Debreu-Farrell assume un valore pari ad 1, ma tali unità non possono essere considerate pienamente efficienti in senso di Koopmans

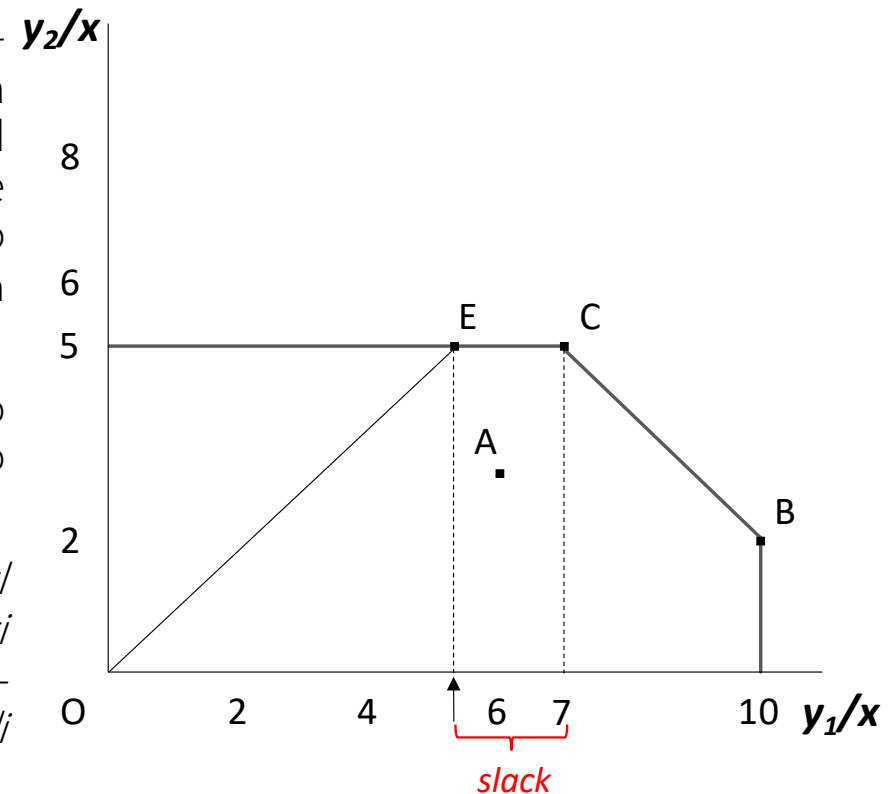


Dov'è lo *slack*? (3/3)

Si prenda in considerazione l'unità E : la sua misura di output-efficienza assume valore pari a 1 ($\overline{OE}/\overline{OE}$), infatti nessuna espansione radiale degli output è possibile continuando ad utilizzare il vettore di input x ; ma un aumento della produzione del solo output y_1 si; infatti, a parità di input, E ha prodotto lo stesso livello di output y_2 prodotto dall'unità produttiva C ma un livello inferiore di output y_1 (*slack*)

Pertanto, l'unità E non è efficiente in senso di Koopmans (con lo stesso livello di input avrebbe potuto produrre un lo stesso livello dell'output y_1 prodotto dall'unità C)

In generale, in corrispondenza di tutte le unità che si collocano sul tratto orizzontale della curva di trasformazione e quelle che si collocano sul tratto verticale della stessa, la misura di output-efficienza di Debreu-Farrell assume un valore pari ad 1, ma tali unità non possono essere considerate efficienti



Modello DEA-CRS *input-oriented* con *slack* (1/6)

- Consideriamo 5 DMU che impiegano due input per produrre un unico output
- I livelli di input per unità output di ciascuna DMU sono riportati nella seguente tabella

DMU	Opex	Capex
T1	1	3
T2	3	2
T3	3.75	1
T4	1.5	1.5
T5	3.9	1

Modello DEA-CRS *input-oriented* con *slack* (2/6)

- Si definiscono:
 - θ la misura di input-efficienza
 - λ_j , con $j = 1, \dots, 5$, i pesi con i quali le DMU reali (osservate) intervengono nelle combinazioni lineari che definiscono i livelli di input della DMU virtuale di riferimento (benchmark)
 - $S_{OPEX}, S_{CAPEX}, S_{RESP}$ gli *slack*
 - ε una quantità molto piccola scelta opportunamente (più piccola di qualsiasi numero positivo)

Modello DEA-CRS *input-oriented* con *slack* (3/6)

- La ricerca del processo-benchmark e la conseguente misura dell'indice di input-efficienza per «T2» può essere condotta mediante la risoluzione del seguente problema di programmazione lineare

$$\begin{aligned} \min \quad & \theta - \varepsilon(S_{OPEX} + S_{CAPEX} + S_{RESP}) \\ \text{s.t.} \quad & 3\theta - S_{OPEX} = \lambda_1 + 3\lambda_2 + 3.75\lambda_3 + 1.5\lambda_4 + 3.9\lambda_5 \\ & 2\theta - S_{CAPEX} = 3\lambda_1 + 2\lambda_2 + \lambda_3 + 1.5\lambda_4 + \lambda_5 \\ & 1 + S_{RESP} = \sum_{j=1}^5 \lambda_j \\ & \lambda_j, S_{OPEX}, S_{CAPEX}, S_{RESP} \geq 0 \\ & 0 < \varepsilon < 1 \end{aligned}$$

Modello DEA-CRS *input-oriented* con *slack* (4/6)

- Soluzione:

Theta	Lambda	Slack
$\theta = 0.6875$	$\lambda_1 = 0$	
	$\lambda_2 = 0$	$S_{OPEX} = 0$
	$\lambda_3 = 0.25$	$S_{CAPEX} = 0$
	$\lambda_4 = 0.75$	$S_{RESP} = 0$
	$\lambda_5 = 0$	

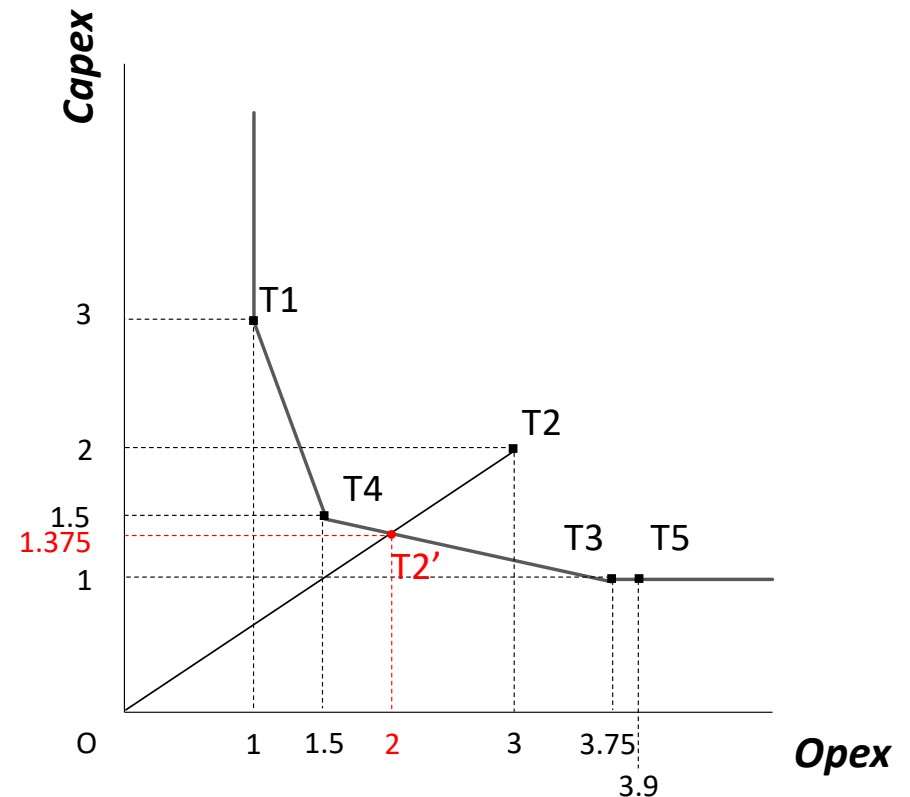
- Interpretazione:

- La DMU «T2» è inefficiente, perché avrebbe potuto realizzare il livello di output y con il 68.75% dei livelli di Opex e Capex impiegati
- Il suo **benchmark** la DMU virtuale che chiameremo «T2'», è una combinazione lineare delle DMU reali «T3» e «T4» (*peers*). Nel dettaglio,

$$x_1^{T2'} = (\lambda_3 \cdot x_1^{T3}) + (\lambda_4 \cdot x_1^{T4}) = (0.25 \cdot 3.75) + (0.75 \cdot 1.5) = 2$$

$$x_2^{T2'} = (\lambda_3 \cdot x_2^{T3}) + (\lambda_4 \cdot x_2^{T4}) = (0.25 \cdot 1) + (0.75 \cdot 1.5) = 1.375$$

- La DMU «T2» presenta *slack* pari a zero



Modello DEA-CRS *input-oriented* con *slack* (5/6)

- La seguente tabella mostra i punteggi di efficienza di tutte le DMU

DMU	θ	λ_1	λ_2	λ_3	λ_4	λ_5	S_OPEX	S_CAPEX	S_RESP
T1	1	1	0	0	0	0	0	0	0
T2	0.6875	0	0	0.25	0.75	0	0	0	0
T3	1	0	0	1	0	0	0	0	0
T4	1	0	0	0	1	0	0	0	0
T5	1	0	0	1	0	0	0.15	0	0

- NB: La DMU T5 ha $\theta = 1$ ma non è pienamente efficiente. Sebbene non sia possibile una contrazione equiproporzionale dei livelli degli input, avrebbe potuto realizzare una unità di output impiegando un livello inferiore di input 1 (Opex). Rispetto al suo benchmark (T3), ha impiegato un eccesso di input 1 pari a 0.15.

Modello DEA-CRS *input-oriented* con *slack* (6/6)

Date K DMU ($j = 1, \dots, K$) ciascuna delle quali produce N output (y_{j1}, \dots, y_{jN}) impiegando M input (x_{j1}, \dots, x_{jM}), consideriamo una generica DMU j_0 . Indicando con:

- λ_j i pesi con i quali le varie unità intervengono nelle combinazioni lineari che definiscono gli input dell'unità virtuale di riferimento
- S_i^- gli input-slack
- S_h^+ gli output-slack
- θ_0 la misura di input-efficienza generica DMU j_0

il modello di programmazione lineare da risolvere per individuare il valore dell'input-efficienza (θ_0) della DMU j_0 e individuare i valori dei pesi (λ_j) è il seguente

$$\begin{aligned}
 \min \quad & \theta_0 - \varepsilon \left[\sum_{i=1}^M S_i^- + \sum_{h=1}^N S_h^+ \right] \\
 \text{s.t.} \quad & \sum_{j=1}^K \lambda_j x_{ij} = \theta_0 x_{ij_0} - S_i^- \quad (i = 1, \dots, M) \\
 & \sum_{j=1}^K \lambda_j y_{hj} = y_{hj_0} + S_h^+ \quad (h = 1, \dots, N) \\
 & \lambda_j, S_i^-, S_h^+ \geq 0 \\
 & 0 < \varepsilon < 1
 \end{aligned}$$

Modello DEA-CRS *output-oriented* con *slack* (1/6)

- Consideriamo 6 DMU che impiegano un input per produrre due output
- I livelli di input e output di ciascuna DMU sono riportati nella seguente tabella

Hospital	Opex (\$000)	No. of severe patients	<i>No. of severe patients per \$ 1,000 Opex</i>	No. of regular patients	<i>No. of regular patients per \$ 1,000 Opex</i>
A	300	900	3	2100	7
B	250	1500	6	1500	6
C	320	2560	8	1280	4
D	400	3200	8	800	2
E	100	500	5	400	4
F	80	480	6	160	2

Modello DEA-CRS *output-oriented* con *slack* (2/6)

- Si definiscono:
 - ϕ una variabile da cui ricavo la misura di output-efficienza
 - $\lambda_A, \lambda_B, \lambda_C, \lambda_D, \lambda_E, \lambda_F$, i pesi con i quali le DMU reali (osservate) intervengono nelle combinazioni lineari che definiscono i livelli di output della DMU virtuale di riferimento (benchmark)
 - $S_{SEVERP}, S_{REGP}, S_{OPEX}$ gli *slack*
 - ε una quantità molto piccola scelta opportunamente (più piccola di qualsiasi numero positivo)

Modello DEA-CRS *output-oriented* con *slack* (3/6)

- La ricerca del processo-benchmark e la conseguente variabile da cui ricavo la misura dell'**indice di output-efficienza** per «E» può essere condotta mediante la risoluzione del seguente problema di programmazione lineare

$$\begin{aligned} \text{Max } & \phi + \varepsilon(S_SEVERP + S_REGP + S_OPEX) \\ \text{S.t. } & 500\phi + S_SEVERP = 900\lambda_A + 1500\lambda_B + 2560\lambda_C + 3200\lambda_D + 500\lambda_E + 480\lambda_F \\ & 400\phi + S_REGP = 2100\lambda_A + 1500\lambda_B + 1280\lambda_C + 800\lambda_D + 400\lambda_E + 160\lambda_F \\ & 100 - S_OPEX = 300\lambda_A + 250\lambda_B + 320\lambda_C + 400\lambda_D + 100\lambda_E + 80\lambda_F \\ & \lambda_j, S_OPEX, S_REGP, S_SEVERP \geq 0 \\ & 0 < \varepsilon < 1 \end{aligned}$$

Modello DEA-CRS *output-oriented* con *slack* (4/6)

- Soluzione:

Phi	Lambda	Slack
$\phi = 1.33$	$\lambda_A = 0$	
	$\lambda_B = 0.67$	$S_{OPEX} = 0$
	$\lambda_C = 0.33$	$S_{SEVERP} = 0$
	$\lambda_D = 0$	$S_{REGP} = 0$
	$\lambda_E = 0$	
	$\lambda_F = 0$	

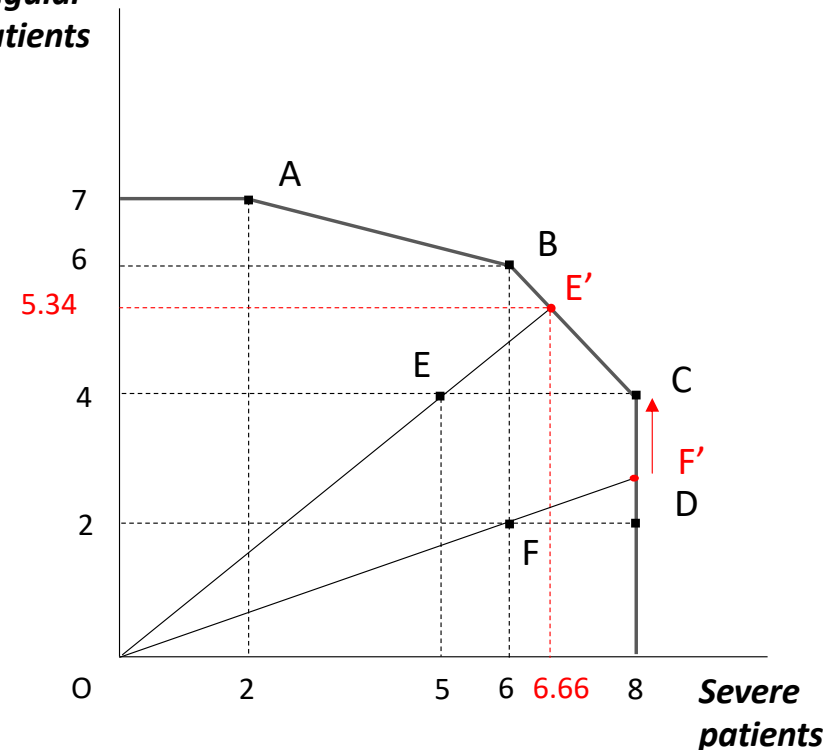
- Interpretazione:

- La DMU «E» è inefficiente (il suo punteggio di output-efficienza è $1/\phi = 0.75$), perché avrebbe potuto realizzare il 33% in più dei livelli di output effettivamente realizzati impiegando lo stesso livello di input
- Il suo **benchmark** la DMU virtuale che chiameremo «E'», è una combinazione lineare delle DMU reali «B» e «C» (*peers*). Nel dettaglio,

$$y_1^{E'} = (\lambda_B \cdot y_1^B) + (\lambda_C \cdot y_1^C) = (0.67 \cdot 6) + (0.33 \cdot 8) = 6.66$$

$$y_2^{E'} = (\lambda_B \cdot y_2^B) + (\lambda_C \cdot y_2^C) = (0.67 \cdot 6) + (0.33 \cdot 4) = 5.34$$
- La DMU «E» presenta *slack* pari a zero

Regular patients



Modello DEA-CRS *output-oriented* con *slack* (5/6)

- La seguente tabella mostra i punteggi di efficienza di tutte le DMU

DMU	ϕ	λ_A	λ_B	λ_C	λ_D	λ_E	λ_F	S_OPEX	S_SEVERP	S_REGP
A	1	1	0	0	0	0	0	0	0	0
B	1	0	1	0	0	0	0	0	0	0
C	1	0	0	1	0	0	0	0	0	0
D	1	0	0	1	0	0	0	0	0	2
E	1.33	0	0.67	0.33	0	0	0	0	0	0
F	1.33	0	0	1	0	0	0	0	0	1.33

- Si osservi che:
 - La DMU «D» ha $\phi = 1$ ma non è pienamente efficiente. Sebbene non sia possibile una espansione equiproporzionale dei livelli degli output, avrebbe potuto realizzare un livello superiore output *REGP* impiegando lo stesso livello di input. Rispetto al suo benchmark («C»), ha prodotto meno output *REGP*
 - La DMU «F» ha $\phi = 1.33$ (cioè, indice di output-efficienza pari a 0.75) pertanto è inefficiente. Per essere considerata efficiente avrebbe dovuto ottenere il 33% in più di entrambi gli output e, inoltre, 1.33 in più del solo output *S_REGP*

Modello DEA-CRS *output-oriented* con *slack* (6/6)

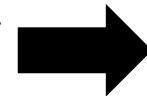
- Date K DMU ($j = 1, \dots, K$) ciascuna delle quali produce N output (y_{j1}, \dots, y_{jN}) impiegando M input (x_{j1}, \dots, x_{jM}), consideriamo una generica DMU j_0 . Indicando con:
- λ_j i pesi con i quali le varie unità intervengono nelle combinazioni lineari che definiscono gli input dell'unità virtuale di riferimento
- S_i^- gli input-slack
- S_h^+ gli output-slack
- ϕ_0 una variabile da cui ricavo la misura di output-efficienza della generica DMU j_0

il modello di programmazione lineare da risolvere per individuare il valore della variabile ϕ_0 e individuare i valori dei pesi λ_j è il seguente:

$$\begin{aligned}
 \max \quad & \phi_0 + \varepsilon \left[\sum_{i=1}^M S_i^- + \sum_{h=1}^N S_h^+ \right] \\
 \text{s.t.} \quad & \sum_{j=1}^K \lambda_j y_{hj} = \phi_0 y_{hj_0} + S_h^+ \quad (h = 1, \dots, N) \\
 & \sum_{j=1}^K \lambda_j x_{ij} = x_{ij_0} - S_i^- \quad (i = 1, \dots, M) \\
 & \lambda_j, S_i^-, S_h^+ \geq 0 \\
 & 0 < \varepsilon < 1
 \end{aligned}$$

Esercizio 3 Modello DEA a rendimenti costanti con slack

- Si realizzi un'analisi DEA con tecnologia a rendimenti costanti di scala (compresa la misurazione degli slack) sul seguente insieme di unità decisionali



DMU	Opex	Capex	y
T1	1	3	1
T2	3	2	1
T3	3.75	1	1
T4	1.5	1.5	1
T5	3.9	1	1

- Rscript

```
x1_es3<-c(1, 3, 3.75, 1.5, 3.9)
x2_es3<-c(3, 2, 1, 1.5, 1)
x_es3<-cbind(x1_es3, x2_es3)
rownames(x_es3)<-c("T1", "T2", "T3", "T4", "T5")
y_es3<-c(1, 1, 1, 1, 1)
y_es3<-cbind(y_es3)
rownames(y_es3)<-c("T1", "T2", "T3", "T4", "T5")
eff_CRS_in_es3<-dea(x_es3, y_es3, RTS="crs", ORIENTATION="in", SLACK=TRUE)
eff(eff_CRS_in_es3)
sl_es3 <- slack(x_es3, y_es3, eff_CRS_in_es3)
data.frame(eff(eff_CRS_in_es3), sl_es3$sx, sl_es3$sy)
peers(eff_CRS_in_es3, NAMES=TRUE)
lambda(eff_CRS_in_es3)
pot_in_rad_sav_es3<-(1-eff(eff_CRS_in_es3)) * x_es3
colnames(pot_in_rad_sav_es3)<-c("Rad_sl_x1", "Rad_sl_x2")
pot_in_rad_sav_es3<-as.data.frame(pot_in_rad_sav_es3)
cal_tot_sl_es3<-data.frame(eff(eff_CRS_in_es3), sl_es3$sx, sl_es3$sy, pot_in_rad_sav_es3)
tot_sl_es3<-data.frame(cal_tot_sl_es3, (cal_tot_sl_es3$sx1 + pot_in_rad_sav_es3$Rad_sl_x1), (cal_tot_sl_es3$sx2 +
pot_in_rad_sav_es3$Rad_sl_x2))
summary(eff_CRS_in_es3)
dea.plot.isoquant(x1_es3,x2_es3,RTS="crs",txt=rownames(x_es3))
```

Esercizio 4 Modello DEA a rendimenti costanti con slack

- Si realizzi un'analisi DEA con tecnologia a rendimenti costanti di scala (compresa la misurazione degli slack) sul seguente insieme di unità decisionali



Hospital	Opex	No. of severe patients per \$ 1,000 Opex	No. of regular patients per \$ 1,000 Opex
A	1000	3	7
B	1000	6	6
C	1000	8	4
D	1000	8	2
E	1000	5	4
F	1000	6	2

- R script

```
x_es4<-c(1000, 1000, 1000, 1000, 1000, 1000)
x_es4<-cbind(x_es4)
rownames(x_es4)<-c("A", "B", "C", "D", "E", "F")
y1_es4<-c(3, 6, 8, 8, 5, 6)
y2_es4<-c(7, 6, 4, 2, 4, 2)
y_es4<-cbind(y1_es4, y2_es4)
rownames(y_es4)<-c("A", "B", "C", "D", "E", "F")
eff_CRS_out_es4<-dea(x_es4, y_es4, RTS="crs", ORIENTATION="out", SLACK=TRUE)
eff(eff_CRS_out_es4)
1/eff(eff_CRS_out_es4)
sl_es4 <- slack(x_es4, y_es4, eff_CRS_out_es4)
data.frame(eff(eff_CRS_out_es4), sl_es4$sx, sl_es4$sy)
peers(eff_CRS_out_es4, NAMES=TRUE)
lambda(eff_CRS_out_es4)
data.frame(eff(eff_CRS_out_es4), lambda(eff_CRS_out_es4))
pot_out_rad_incr_es4<-((eff_CRS_out_es4$eff)*y_es4) - y_es4
colnames(pot_out_rad_incr_es4)<-c("Rad_sl_y1", "Rad_sl_y2")
pot_out_rad_incr_es4<-as.data.frame(pot_out_rad_incr_es4)
cal_tot_sl_es4<-data.frame(eff(eff_CRS_out_es4), sl_es4$sx, sl_es4$sy, pot_out_rad_incr_es4)
tot_sl_es4<-data.frame(cal_tot_sl_es4, (cal_tot_sl_es4$sy1 + pot_out_rad_incr_es4$Rad_sl_y1), (cal_tot_sl_es4$sy2 +
pot_out_rad_incr_es4$Rad_sl_y2))
summary(eff_CRS_out_es4)
dea.plot.transform(y1_es4,y2_es4,RTS="crs",txt=rownames(y_es4))
```

Modelli DEA VRS con *slack*

Modelli DEA-VRS (1/7)

- Consideriamo 5 DMU che impiegano un unico input per produrre un unico output
- I livelli di input e output di ciascuna DMU sono riportati nella seguente tabella

DMU	x	y
S1	4	4
S2	5	8
S3	3	7
S4	6	10
S5	2	2

Modelli DEA-VRS (2/7)

Costruiamo l'insieme di produzione assumendo le ipotesi 1-4

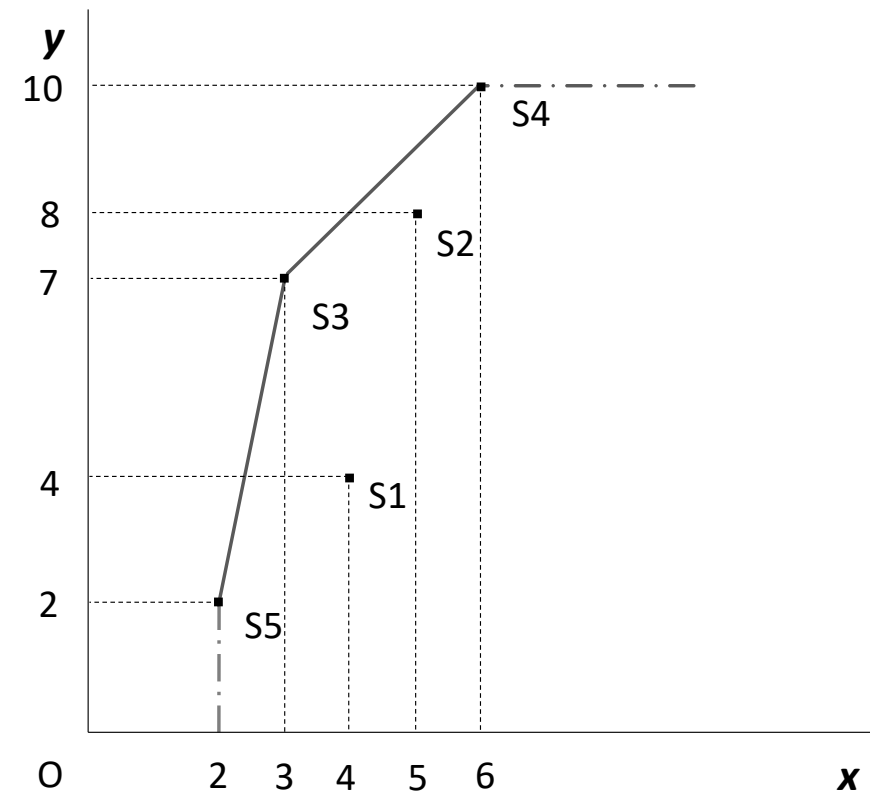
I1: I processi produttivi osservati appartengono a Z

I2: Ipotesi di *free disposa*/per gli output

I3: Ipotesi di *free disposa*/per gli input

I4: Convessità

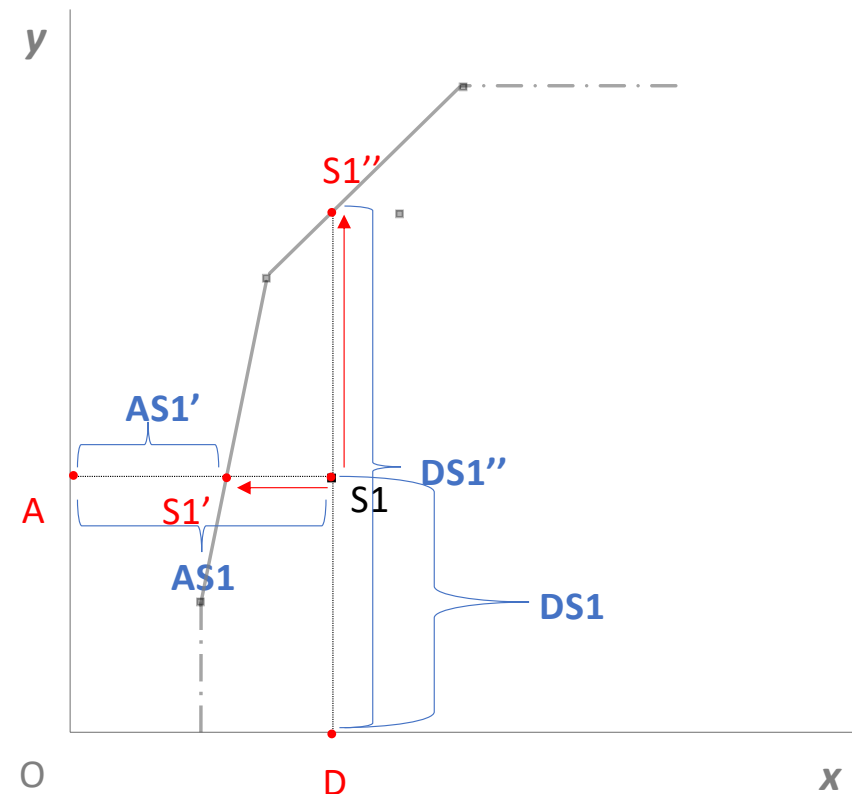
Il confine dell'insieme di produzione è la frontiera spezzata raffigurata in questa slide



Modelli DEA-VRS (3/7)

- Misuriamo l'efficienza della DMU S1
 - Graficamente:
 - Input-efficienza $AS1'/AS1$
 - Output-efficienza $DS1''/DS1^{(*)}$
 - La ricerca dei processi-benchmark e le conseguenti misure di input-efficienza e output-efficienza della DMU S1 può essere condotta mediante la risoluzione dei seguenti modelli di programmazione lineare...

(*) Per ottenere una misura compresa tra 0 e 1, si considera il suo reciproco



Modelli DEA-VRS (4/7)

$j = 1, \dots, K \rightarrow$ DMUs $y_{j1}, \dots, y_{jN} \rightarrow$ outputs $x_{j1}, \dots, x_{jM} \rightarrow$ inputs $\lambda_j \rightarrow$ pesi $S_i^- \rightarrow$ input-slacks $S_h^+ \rightarrow$ output-slacks	$\theta \rightarrow$ la misura di input-efficienza $\phi_0 \rightarrow$ una variabile da cui ricavo la misura di output-efficienza della generica DMU j_0 $\varepsilon \rightarrow$ una quantità molto piccola scelta opportunamente (più piccola di qualsiasi numero positivo)
--	---

Modello DEA-VRS *input-oriented*

$$\begin{aligned}
 \min \quad & \theta_0 - \varepsilon \left[\sum_{i=1}^M S_i^- + \sum_{h=1}^N S_h^+ \right] \\
 \text{s.t.} \quad & \sum_{j=1}^K \lambda_j x_{ij} = \theta_0 x_{ij_0} - S_i^- \quad (i = 1, \dots, M) \\
 & \sum_{j=1}^K \lambda_j y_{hj} = y_{hj_0} + S_h^+ \quad (h = 1, \dots, N) \\
 & \sum_{j=1}^K \lambda_j = 1 \\
 & \lambda_j, S_i^-, S_h^+ \geq 0 \\
 & 0 < \varepsilon < 1
 \end{aligned}$$

Modello DEA-VRS *output-oriented*

$$\begin{aligned}
 \max \quad & \phi_0 + \varepsilon \left[\sum_{i=1}^M S_i^- + \sum_{h=1}^N S_h^+ \right] \\
 \text{s.t.} \quad & \sum_{j=1}^K \lambda_j y_{hj} = \phi_0 y_{hj_0} + S_h^+ \quad (h = 1, \dots, N) \\
 & \sum_{j=1}^K \lambda_j x_{ij} = x_{ij_0} - S_i^- \quad (i = 1, \dots, M) \\
 & \sum_{j=1}^K \lambda_j = 1 \\
 & \lambda_j, S_i^-, S_h^+ \geq 0 \\
 & 0 < \varepsilon < 1
 \end{aligned}$$

Modelli DEA-VRS (5/7)

Risultati DEA-VRS *input-oriented*

DMU	θ	λ_3	λ_4	λ_5	S_i	S_h
S1	0.6	0.4	0	0.6	0	0
S2	0.8	0.67	0.33	0	0	0
S3	1	1	0	0	0	0
S4	1	0	1	0	0	0
S5	1	0	0	1	0	0

Risultati DEA-VRS *output-oriented*

DMU	ϕ	λ_3	λ_4	λ_5	S_i	S_h
S1	2	0.67	0.33	0	0	0
S2	1.12	0.33	0.67	0	0	0
S3	1	1	0	0	0	0
S4	1	0	1	0	0	0
S5	1	0	0	1	0	0

Modelli DEA-VRS (6/7)

Interpretazione risultati DEA-VRS *input-oriented* – DMU S1

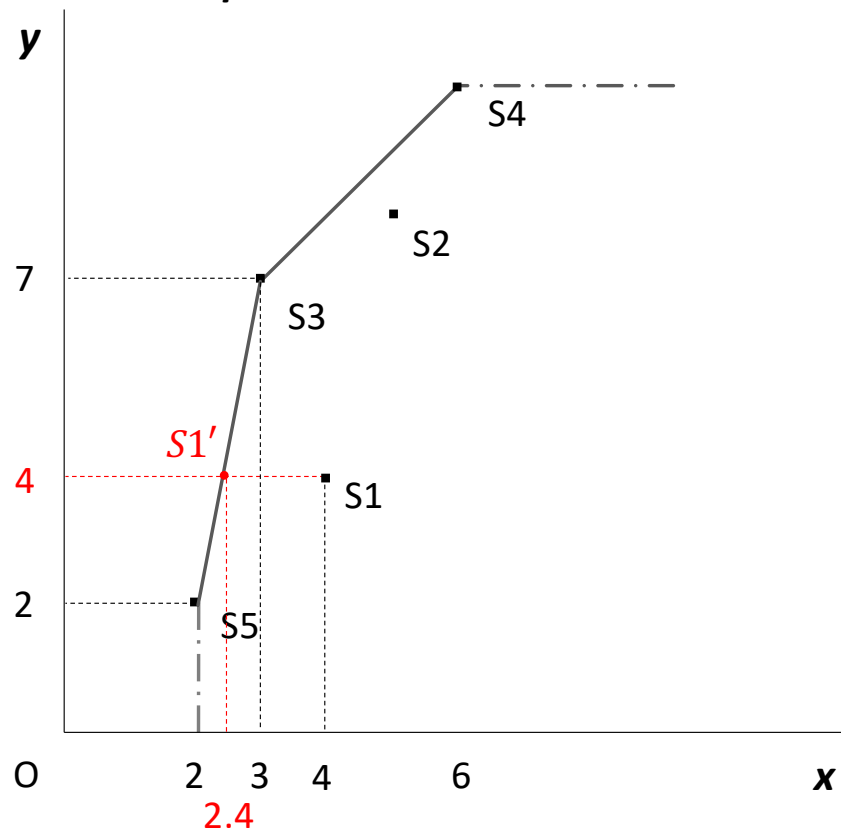
- In merito alla DMU S1 si osserva che:
 - Indice di input-efficienza
 $\theta = 0.6$
 - Benchmark
$$\left. \begin{aligned} x^{S1'} &= (0.4 \cdot 3) + (0.6 \cdot 2) = 2.4 \\ y^{S1'} &= (0.4 \cdot 7) + (0.6 \cdot 2) = 4 \end{aligned} \right\} S1'(2.4,4)$$
 - Peers
 - S3
 - S5

Interpretazione risultati DEA-VRS *output-oriented* – DMU S1

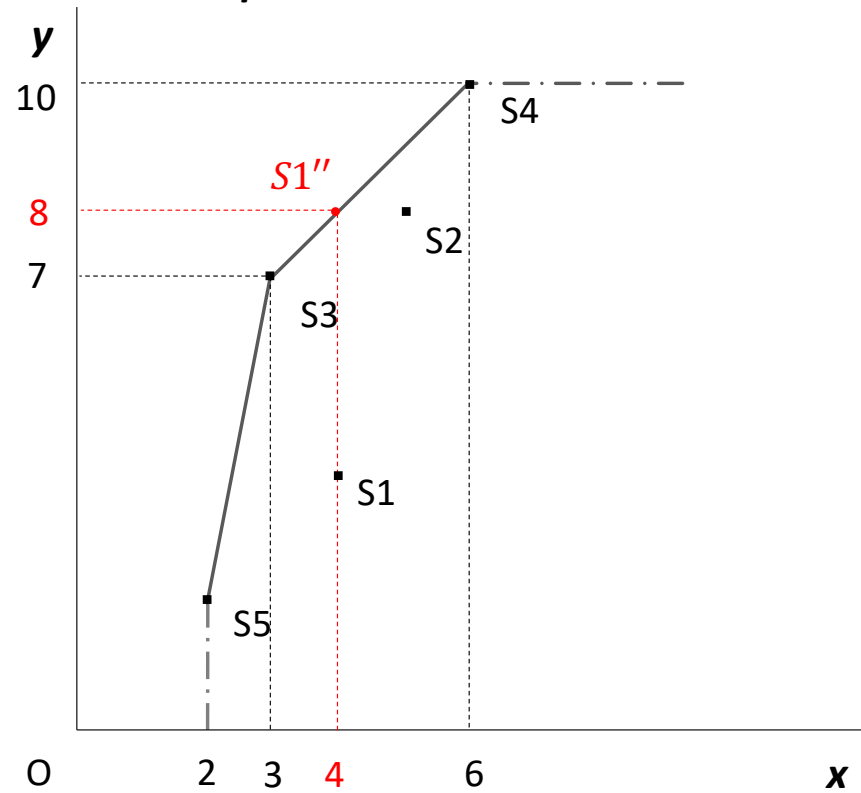
- In merito alla DMU S1 si osserva che:
 - Indice di output-efficienza
 $1/\phi = 0.5$
 - Benchmark
$$\left. \begin{aligned} x^{S1''} &= (0.67 \cdot 3) + (0.33 \cdot 6) = 4 \\ y^{S1''} &= (0.67 \cdot 7) + (0.33 \cdot 10) = 8 \end{aligned} \right\} S1''(4,8)$$
 - Peers
 - S3
 - S4

Modelli DEA-VRS (7/7)

Rappresentazione grafica risultati DEA-VRS
input-oriented – DMU S1

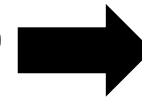


Rappresentazione grafica risultati DEA-VRS
output-oriented – DMU S1



Esercizio 5 Modello DEA a rendimenti variabili con slack

- Si realizzi un'analisi DEA con tecnologia a rendimenti variabili di scala (compresa la misurazione degli slack) sul seguente insieme di unità decisionali



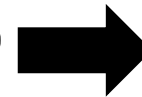
DMU	x	y
S1	4	4
S2	5	8
S3	3	7
S4	6	10
S5	2	2

- R script

```
x_es5<-c(4, 5, 3, 6, 2)
x_es5<-cbind(x_es5)
rownames(x_es5)<-c("S1", "S2", "S3", "S4", "S5")
y_es5<-c(4, 8, 7, 10, 2)
y_es5<-cbind(y_es5)
rownames(y_es5)<-c("S1", "S2", "S3", "S4", "S5")
eff_VRS_in_es5<-dea(x_es5, y_es5, RTS="vrs", ORIENTATION="in", SLACK=TRUE)
eff(eff_VRS_in_es5)
sl_es5 <- slack(x_es5, y_es5, eff_VRS_in_es5)
data.frame(eff(eff_VRS_in_es5), sl_es5$sx, sl_es5$sy)
peers(eff_VRS_in_es5, NAMES=TRUE)
lambda(eff_VRS_in_es5)
data.frame(eff(eff_VRS_in_es5), lambda(eff_VRS_in_es5))
pot_in_rad_sav_es5<-(1-eff(eff_VRS_in_es5)) * x_es5
colnames(pot_in_rad_sav_es5)<-c("Rad_sl_x1")
pot_in_rad_sav_es5<-as.data.frame(pot_in_rad_sav_es5)
cal_tot_sl_es5<-data.frame(eff(eff_VRS_in_es5), sl_es5$sx, sl_es5$sy, pot_in_rad_sav_es5)
tot_sl_es5<-data.frame(cal_tot_sl_es5, (cal_tot_sl_es5$sx1 + pot_in_rad_sav_es5$Rad_sl_x1))
summary(eff_VRS_in_es5)
dea.plot.frontier(x_es5,y_es5,RTS="vrs",txt=rownames(x_es5))
```

Esercizio 6 Modello DEA a rendimenti variabili con slack

- Si realizzi un'analisi DEA con tecnologia a rendimenti variabili di scala (compresa la misurazione degli slack) sul seguente insieme di unità decisionali



DMU	x	y
S1	4	4
S2	5	8
S3	3	7
S4	6	10
S5	2	2

- R script

```
x_es5<-c(4, 5, 3, 6, 2)
x_es5<-cbind(x_es5)
rownames(x_es5)<-c("S1", "S2", "S3", "S4", "S5")
y_es5<-c(4, 8, 7, 10, 2)
y_es5<-cbind(y_es5)
rownames(y_es5)<-c("S1", "S2", "S3", "S4", "S5")
eff_VRS_out_es6<-dea(x_es5, y_es5, RTS="vrs", ORIENTATION="out", SLACK=TRUE)
eff(eff_VRS_out_es6)
1/eff(eff_VRS_out_es6)
sl_es6 <- slack(x_es5, y_es5, eff_VRS_out_es6)
data.frame(eff(eff_VRS_out_es6), sl_es6$sx, sl_es6$sy)
peers(eff_VRS_out_es6, NAMES=TRUE)
lambda(eff_VRS_out_es6)
data.frame(eff(eff_VRS_out_es6), lambda(eff_VRS_out_es6))
pot_out_rad_incr_es6<-((eff_VRS_out_es6$eff)*y_es5) - y_es5
colnames(pot_out_rad_incr_es6)<-c("Rad_sl_y1")
pot_out_rad_incr_es6<-as.data.frame(pot_out_rad_incr_es6)
cal_tot_sl_es6<-data.frame(eff(eff_VRS_out_es6), sl_es6$sx, sl_es6$sy, pot_out_rad_incr_es6)
tot_sl_es6<-data.frame(cal_tot_sl_es6, (cal_tot_sl_es6$sy1 + pot_out_rad_incr_es6$Rad_sl_y1))
summary(eff_VRS_out_es6)
```

Grazie per l'attenzione!

Marialisa Mazzocchitti

Mail: m.mazzocchitti@unich.it

Skype: marialisa_mazzocchitti