QMM Assignment 5 Module 8

YASH BHANUSHALI

2022-10-30

## Question 1 - Hope Valley Health Care Association

Problem Description – The Hope Valley Health Care Association owns and operates six nursing homes in adjoining states. An evaluation of their efficiency has been undertaken using two inputs and two outputs. The inputs are staffing labor (measured in average hours per day) and the cost of supplies (in thousands of dollars per day). The outputs are the number of patient-days reimbursed by third-party sources and the number of patient-days reimbursed privately. A summary of performance data is shown in the table below.

# This package is required for running the DEA functions in this program  
require(Benchmarking)

## Loading required package: Benchmarking

## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog

##   
## Loading Benchmarking version 0.30h, (Revision 244, 2022/05/05 16:31:31) ...

## Build 2022/05/05 16:31:40

Next, the problem data will be loaded into the R environment.

# Create matrix for the two inputs  
X <- matrix(c(150, 400, 320, 520, 350, 320, 0.2, 0.7, 1.2, 2.0, 1.2, 0.7), ncol = 2)  
# Create matrix for the two outputs  
Y <- matrix(c(14000, 14000, 42000, 28000, 19000, 14000, 3500, 21000, 10500, 42000, 25000, 15000), ncol = 2)  
# Name the columns of the inputs and outputs  
colnames(X) <- c("Staff Hours per Day","Supplies per Day")  
colnames(Y) <- c("Reimburse Patient-Days", "Privately Paid Patient-Days")  
# Return the matrices for review  
print(X)

## Staff Hours per Day Supplies per Day  
## [1,] 150 0.2  
## [2,] 400 0.7  
## [3,] 320 1.2  
## [4,] 520 2.0  
## [5,] 350 1.2  
## [6,] 320 0.7

print(Y)

## Reimburse Patient-Days Privately Paid Patient-Days  
## [1,] 14000 3500  
## [2,] 14000 21000  
## [3,] 42000 10500  
## [4,] 28000 42000  
## [5,] 19000 25000  
## [6,] 14000 15000

The results of DEA using the FDH method will be returned by the next chunch of code.

# DEA code utilizing the FDH method  
FDH <- rep("FDH", times = 6)  
Not\_Applicable <- rep(NA, times = 6)  
DEA\_FDH <- dea(X, Y, RTS = "FDH")  
DEA\_FDH\_Peers <- peers(DEA\_FDH)   
DEA\_FDH\_Lambda <- lambda(DEA\_FDH)   
print(DEA\_FDH)

## [1] 1 1 1 1 1 1

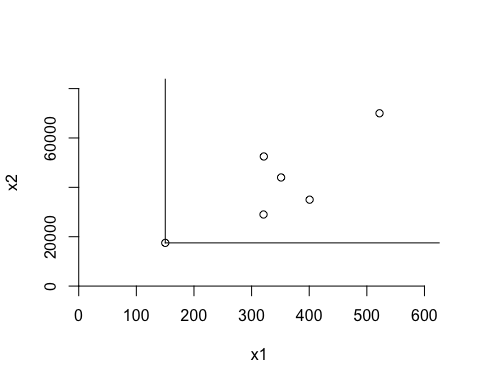
print(DEA\_FDH\_Peers)

## peer1  
## [1,] 1  
## [2,] 2  
## [3,] 3  
## [4,] 4  
## [5,] 5  
## [6,] 6

print(DEA\_FDH\_Lambda)

## L1 L2 L3 L4 L5 L6  
## [1,] 1 0 0 0 0 0  
## [2,] 0 1 0 0 0 0  
## [3,] 0 0 1 0 0 0  
## [4,] 0 0 0 1 0 0  
## [5,] 0 0 0 0 1 0  
## [6,] 0 0 0 0 0 1

dea.plot.isoquant(X, Y, RTS= "FDH")



# Summarize the results for addition to a summary table  
DEA\_FDH\_Peers <- cbind(DEA\_FDH\_Peers, Not\_Applicable, Not\_Applicable)  
FDH\_Summary <- cbind(FDH, DEA\_FDH$eff, DEA\_FDH\_Peers, DEA\_FDH\_Lambda)  
colnames(FDH\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
print(FDH\_Summary)

## Method Eff P1 P2 P3 L1 L2 L3 L4 L5 L6   
## [1,] "FDH" "1" "1" NA NA "1" "0" "0" "0" "0" "0"  
## [2,] "FDH" "1" "2" NA NA "0" "1" "0" "0" "0" "0"  
## [3,] "FDH" "1" "3" NA NA "0" "0" "1" "0" "0" "0"  
## [4,] "FDH" "1" "4" NA NA "0" "0" "0" "1" "0" "0"  
## [5,] "FDH" "1" "5" NA NA "0" "0" "0" "0" "1" "0"  
## [6,] "FDH" "1" "6" NA NA "0" "0" "0" "0" "0" "1"

The results of DEA using the CRS method will be returned by the next chunch of code.

# DEA code utilizing the CRS method  
CRS <- rep("CRS", times = 6)  
DEA\_CRS <- dea(X, Y, RTS = "CRS")  
DEA\_CRS\_Peers <- peers(DEA\_CRS)   
DEA\_CRS\_Lambda <- lambda(DEA\_CRS)   
print(DEA\_CRS)

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

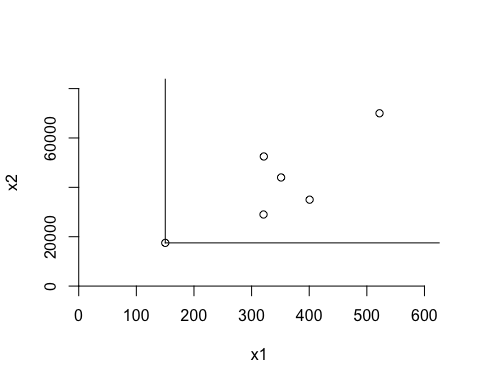
print(DEA\_CRS\_Peers)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 1 2 4  
## [6,] 1 2 4

print(DEA\_CRS\_Lambda)

## L1 L2 L3 L4  
## [1,] 1.0000000 0.00000000 0 0.0000000  
## [2,] 0.0000000 1.00000000 0 0.0000000  
## [3,] 0.0000000 0.00000000 1 0.0000000  
## [4,] 0.0000000 0.00000000 0 1.0000000  
## [5,] 0.2000000 0.08048142 0 0.5383307  
## [6,] 0.3428571 0.39499264 0 0.1310751

dea.plot.isoquant(X, Y, RTS= "CRS")



# Summarize the results for addition to a summary table  
DEA\_CRS\_Lambda <- cbind(DEA\_CRS\_Lambda, Not\_Applicable, Not\_Applicable)  
CRS\_Summary <- cbind(CRS, DEA\_CRS$eff, DEA\_CRS\_Peers, DEA\_CRS\_Lambda)  
colnames(CRS\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
CRS\_Summary <- as.data.frame(CRS\_Summary)  
CRS\_Summary

## Method Eff P1 P2 P3 L1 L2 L3  
## 1 CRS 1 1 <NA> <NA> 1 0 0  
## 2 CRS 1 2 <NA> <NA> 0 1 0  
## 3 CRS 1 3 <NA> <NA> 0 0 1  
## 4 CRS 1 4 <NA> <NA> 0 0 0  
## 5 CRS 0.977498691784406 1 2 4 0.2 0.0804814233385661 0  
## 6 CRS 0.867452135493373 1 2 4 0.342857142857143 0.39499263622975 0  
## L4 L5 L6  
## 1 0 <NA> <NA>  
## 2 0 <NA> <NA>  
## 3 0 <NA> <NA>  
## 4 1 <NA> <NA>  
## 5 0.538330716902146 <NA> <NA>  
## 6 0.131075110456554 <NA> <NA>

The results of DEA using the VRS method will be returned by the next chunch of code.

# DEA code utilizing the VRS method  
VRS <- rep("VRS", times = 6)  
DEA\_VRS <- dea(X, Y, RTS = "VRS")  
DEA\_VRS\_Peers <- peers(DEA\_VRS)   
DEA\_VRS\_Lambda <- lambda(DEA\_VRS)   
print(DEA\_VRS)

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

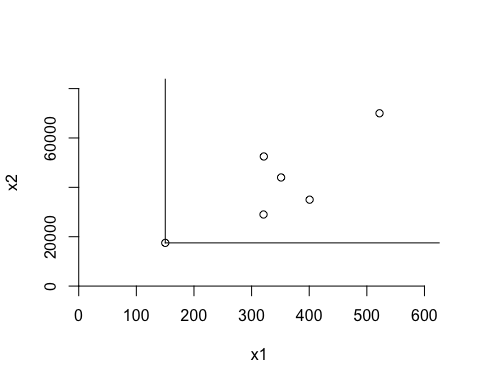
print(DEA\_VRS\_Peers)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 5 NA NA  
## [6,] 1 2 5

print(DEA\_VRS\_Lambda)

## L1 L2 L3 L4 L5  
## [1,] 1.0000000 0.0000000 0 0 0.0000000  
## [2,] 0.0000000 1.0000000 0 0 0.0000000  
## [3,] 0.0000000 0.0000000 1 0 0.0000000  
## [4,] 0.0000000 0.0000000 0 1 0.0000000  
## [5,] 0.0000000 0.0000000 0 0 1.0000000  
## [6,] 0.4014399 0.3422606 0 0 0.2562995

dea.plot.isoquant(X, Y, RTS= "VRS")



# Summarize the results for addition to a summary table  
DEA\_VRS\_Lambda <- cbind(DEA\_VRS\_Lambda, Not\_Applicable)  
VRS\_Summary <- cbind(VRS, DEA\_VRS$eff, DEA\_VRS\_Peers, DEA\_VRS\_Lambda)  
colnames(VRS\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
VRS\_Summary <- as.data.frame(VRS\_Summary)  
VRS\_Summary

## Method Eff P1 P2 P3 L1 L2 L3  
## 1 VRS 1 1 <NA> <NA> 1 0 0  
## 2 VRS 1 2 <NA> <NA> 0 1 0  
## 3 VRS 1 3 <NA> <NA> 0 0 1  
## 4 VRS 1 4 <NA> <NA> 0 0 0  
## 5 VRS 1 5 <NA> <NA> 0 0 0  
## 6 VRS 0.896328293736501 1 2 5 0.401439884809215 0.342260619150468 0  
## L4 L5 L6  
## 1 0 0 <NA>  
## 2 0 0 <NA>  
## 3 0 0 <NA>  
## 4 1 0 <NA>  
## 5 0 1 <NA>  
## 6 0 0.256299496040317 <NA>

The results of DEA using the IRS approach will be returned by the next chunch of code.

# DEA code utilizing the IRS method  
IRS <- rep("IRS", times = 6)  
DEA\_IRS <- dea(X, Y, RTS = "IRS")  
DEA\_IRS\_Peers <- peers(DEA\_IRS)   
DEA\_IRS\_Lambda <- lambda(DEA\_IRS)   
print(DEA\_IRS)

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

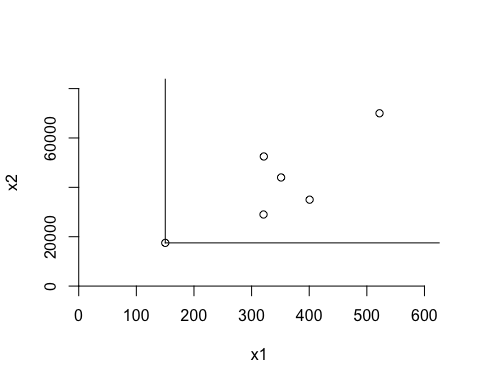
print(DEA\_IRS\_Peers)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 5 NA NA  
## [6,] 1 2 5

print(DEA\_IRS\_Lambda)

## L1 L2 L3 L4 L5  
## [1,] 1.0000000 0.0000000 0 0 0.0000000  
## [2,] 0.0000000 1.0000000 0 0 0.0000000  
## [3,] 0.0000000 0.0000000 1 0 0.0000000  
## [4,] 0.0000000 0.0000000 0 1 0.0000000  
## [5,] 0.0000000 0.0000000 0 0 1.0000000  
## [6,] 0.4014399 0.3422606 0 0 0.2562995

dea.plot.isoquant(X, Y, RTS= "IRS")



# Summarize the results for addition to a summary table  
DEA\_IRS\_Lambda <- cbind(DEA\_IRS\_Lambda, Not\_Applicable)  
IRS\_Summary <- cbind(IRS, DEA\_IRS$eff, DEA\_IRS\_Peers, DEA\_IRS\_Lambda)  
colnames(IRS\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
IRS\_Summary <- as.data.frame(IRS\_Summary)  
IRS\_Summary

## Method Eff P1 P2 P3 L1 L2 L3  
## 1 IRS 1 1 <NA> <NA> 1 0 0  
## 2 IRS 1 2 <NA> <NA> 0 1 0  
## 3 IRS 1 3 <NA> <NA> 0 0 1  
## 4 IRS 1 4 <NA> <NA> 0 0 0  
## 5 IRS 1 5 <NA> <NA> 0 0 0  
## 6 IRS 0.896328293736501 1 2 5 0.401439884809215 0.342260619150468 0  
## L4 L5 L6  
## 1 0 0 <NA>  
## 2 0 0 <NA>  
## 3 0 0 <NA>  
## 4 1 0 <NA>  
## 5 0 1 <NA>  
## 6 0 0.256299496040317 <NA>

The results of DEA using the DRS method will be returned by the next section of code.

# DEA code utilizing the DRS method  
DRS <- rep("DRS", times = 6)  
DEA\_DRS <- dea(X, Y, RTS = "DRS")  
DEA\_DRS\_Peers <- peers(DEA\_DRS)   
DEA\_DRS\_Lambda <- lambda(DEA\_DRS)   
print(DEA\_DRS)

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

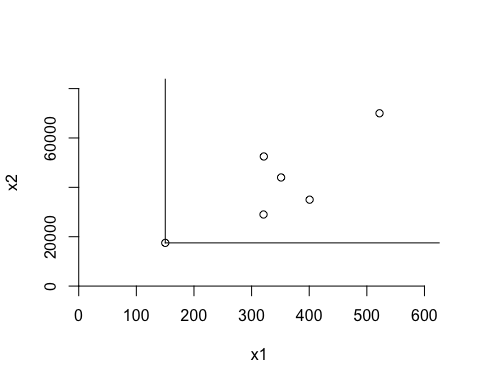
print(DEA\_DRS\_Peers)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 1 2 4  
## [6,] 1 2 4

print(DEA\_DRS\_Lambda)

## L1 L2 L3 L4  
## [1,] 1.0000000 0.00000000 0 0.0000000  
## [2,] 0.0000000 1.00000000 0 0.0000000  
## [3,] 0.0000000 0.00000000 1 0.0000000  
## [4,] 0.0000000 0.00000000 0 1.0000000  
## [5,] 0.2000000 0.08048142 0 0.5383307  
## [6,] 0.3428571 0.39499264 0 0.1310751

dea.plot.isoquant(X, Y, RTS= "DRS")



# Summarize the results for addition to a summary table  
DEA\_DRS\_Lambda <- cbind(DEA\_DRS\_Lambda, Not\_Applicable, Not\_Applicable)  
DRS\_Summary <- cbind(DRS, DEA\_DRS$eff, DEA\_DRS\_Peers, DEA\_DRS\_Lambda)  
colnames(DRS\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
DRS\_Summary <- as.data.frame(DRS\_Summary)  
DRS\_Summary

## Method Eff P1 P2 P3 L1 L2 L3  
## 1 DRS 1 1 <NA> <NA> 1 0 0  
## 2 DRS 1 2 <NA> <NA> 0 1 0  
## 3 DRS 1 3 <NA> <NA> 0 0 1  
## 4 DRS 1 4 <NA> <NA> 0 0 0  
## 5 DRS 0.977498691784406 1 2 4 0.2 0.0804814233385655 0  
## 6 DRS 0.867452135493373 1 2 4 0.342857142857143 0.394992636229749 0  
## L4 L5 L6  
## 1 0 <NA> <NA>  
## 2 0 <NA> <NA>  
## 3 0 <NA> <NA>  
## 4 1 <NA> <NA>  
## 5 0.538330716902146 <NA> <NA>  
## 6 0.131075110456554 <NA> <NA>

The results of DEA using the FRH/ADD technique will be returned by the following chunch of code.

# DEA code utilizing the ADD method  
ADD <- rep("ADD", times = 6)  
DEA\_ADD <- dea(X, Y, RTS = "ADD")  
DEA\_ADD\_Peers <- peers(DEA\_ADD)   
DEA\_ADD\_Lambda <- lambda(DEA\_ADD)   
print(DEA\_ADD)

## [1] 1 1 1 1 1 1

print(DEA\_ADD\_Peers)

## peer1  
## [1,] 1  
## [2,] 2  
## [3,] 3  
## [4,] 4  
## [5,] 5  
## [6,] 6

print(DEA\_ADD\_Lambda)

## L1 L2 L3 L4 L5 L6  
## [1,] 1 0 0 0 0 0  
## [2,] 0 1 0 0 0 0  
## [3,] 0 0 1 0 0 0  
## [4,] 0 0 0 1 0 0  
## [5,] 0 0 0 0 1 0  
## [6,] 0 0 0 0 0 1

# Summarize the results for addition to a summary table  
DEA\_ADD\_Peers <- cbind(DEA\_ADD\_Peers, Not\_Applicable, Not\_Applicable)  
ADD\_Summary <- cbind(ADD, DEA\_ADD$eff, DEA\_ADD\_Peers, DEA\_ADD\_Lambda)  
colnames(ADD\_Summary) <- c("Method","Eff", "P1", "P2", "P3", "L1", "L2", "L3", "L4", "L5", "L6")  
ADD\_Summary <- as.data.frame(ADD\_Summary)  
ADD\_Summary

## Method Eff P1 P2 P3 L1 L2 L3 L4 L5 L6  
## 1 ADD 1 1 <NA> <NA> 1 0 0 0 0 0  
## 2 ADD 1 2 <NA> <NA> 0 1 0 0 0 0  
## 3 ADD 1 3 <NA> <NA> 0 0 1 0 0 0  
## 4 ADD 1 4 <NA> <NA> 0 0 0 1 0 0  
## 5 ADD 1 5 <NA> <NA> 0 0 0 0 1 0  
## 6 ADD 1 6 <NA> <NA> 0 0 0 0 0 1

# Combine all of the method summary tables into one large summary table for each method  
Summary\_Table <- rbind(FDH\_Summary, CRS\_Summary, VRS\_Summary, IRS\_Summary, DRS\_Summary, ADD\_Summary)  
# Return the summary table for review  
print(Summary\_Table)

## Method Eff P1 P2 P3 L1 L2  
## 1 FDH 1 1 <NA> <NA> 1 0  
## 2 FDH 1 2 <NA> <NA> 0 1  
## 3 FDH 1 3 <NA> <NA> 0 0  
## 4 FDH 1 4 <NA> <NA> 0 0  
## 5 FDH 1 5 <NA> <NA> 0 0  
## 6 FDH 1 6 <NA> <NA> 0 0  
## 7 CRS 1 1 <NA> <NA> 1 0  
## 8 CRS 1 2 <NA> <NA> 0 1  
## 9 CRS 1 3 <NA> <NA> 0 0  
## 10 CRS 1 4 <NA> <NA> 0 0  
## 11 CRS 0.977498691784406 1 2 4 0.2 0.0804814233385661  
## 12 CRS 0.867452135493373 1 2 4 0.342857142857143 0.39499263622975  
## 13 VRS 1 1 <NA> <NA> 1 0  
## 14 VRS 1 2 <NA> <NA> 0 1  
## 15 VRS 1 3 <NA> <NA> 0 0  
## 16 VRS 1 4 <NA> <NA> 0 0  
## 17 VRS 1 5 <NA> <NA> 0 0  
## 18 VRS 0.896328293736501 1 2 5 0.401439884809215 0.342260619150468  
## 19 IRS 1 1 <NA> <NA> 1 0  
## 20 IRS 1 2 <NA> <NA> 0 1  
## 21 IRS 1 3 <NA> <NA> 0 0  
## 22 IRS 1 4 <NA> <NA> 0 0  
## 23 IRS 1 5 <NA> <NA> 0 0  
## 24 IRS 0.896328293736501 1 2 5 0.401439884809215 0.342260619150468  
## 25 DRS 1 1 <NA> <NA> 1 0  
## 26 DRS 1 2 <NA> <NA> 0 1  
## 27 DRS 1 3 <NA> <NA> 0 0  
## 28 DRS 1 4 <NA> <NA> 0 0  
## 29 DRS 0.977498691784406 1 2 4 0.2 0.0804814233385655  
## 30 DRS 0.867452135493373 1 2 4 0.342857142857143 0.394992636229749  
## 31 ADD 1 1 <NA> <NA> 1 0  
## 32 ADD 1 2 <NA> <NA> 0 1  
## 33 ADD 1 3 <NA> <NA> 0 0  
## 34 ADD 1 4 <NA> <NA> 0 0  
## 35 ADD 1 5 <NA> <NA> 0 0  
## 36 ADD 1 6 <NA> <NA> 0 0  
## L3 L4 L5 L6  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 1 0 0 0  
## 4 0 1 0 0  
## 5 0 0 1 0  
## 6 0 0 0 1  
## 7 0 0 <NA> <NA>  
## 8 0 0 <NA> <NA>  
## 9 1 0 <NA> <NA>  
## 10 0 1 <NA> <NA>  
## 11 0 0.538330716902146 <NA> <NA>  
## 12 0 0.131075110456554 <NA> <NA>  
## 13 0 0 0 <NA>  
## 14 0 0 0 <NA>  
## 15 1 0 0 <NA>  
## 16 0 1 0 <NA>  
## 17 0 0 1 <NA>  
## 18 0 0 0.256299496040317 <NA>  
## 19 0 0 0 <NA>  
## 20 0 0 0 <NA>  
## 21 1 0 0 <NA>  
## 22 0 1 0 <NA>  
## 23 0 0 1 <NA>  
## 24 0 0 0.256299496040317 <NA>  
## 25 0 0 <NA> <NA>  
## 26 0 0 <NA> <NA>  
## 27 1 0 <NA> <NA>  
## 28 0 1 <NA> <NA>  
## 29 0 0.538330716902146 <NA> <NA>  
## 30 0 0.131075110456554 <NA> <NA>  
## 31 0 0 0 0  
## 32 0 0 0 0  
## 33 1 0 0 0  
## 34 0 1 0 0  
## 35 0 0 1 0  
## 36 0 0 0 1

The summary table reveals that the FRH and FDH algorithms both return efficiency of 1.0, the same peer and lambda values, and this for all six DMUs. DMU[1:4] was discovered to be effective at 1.0 by the CRS approach. DMU[1:5] was discovered to be effective at 1.0 through the VRS approach. DMU[1:4] was shown to be efficient at 1.0 using the DRS technique and IRS, respectively. All of the less efficient DMUs had Peer[1] and Peer[2] values of 1 and 2, respectively; the Peer[3] value, however, varied depending on the method, being either 4 or 5. Furthermore, for the identical DMU, the relative weights (lambdas) for all approaches were rather close.