MIS 64018: Assignment_5: DEA Problem formulation and Goal Programming

Eyob Tadele

11/05/2021

Project Objective

The purpose of this assignment is to:
* Explore the use of DEA, formulate and solve DEA problems under different assumptions.
* Compare and contrast these results
* Explore goal programming formulations and solutions

Question 1: 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.

DMU	St Hrs/day	SS/day	Reim Pat-Days	PvtP Pat-Days
Facility 1	150	0.2	14000	3500
Facility 2	400	0.7	14000	21000
Facility 3	320	1.2	42000	10500
Facility 4	520	2.0	28000	42000
Facility 5	350	1.2	19000	25000
Facility 6	320	0.7	14000	15000

Problem Formulation:

```
library(lpSolveAPI)
dmu1 <- read.lp("DMU1.lp")
dmu2 <- read.lp("DMU2.lp")
dmu3 <- read.lp("DMU3.lp")
dmu4 <- read.lp("DMU4.lp")
dmu5 <- read.lp("DMU5.lp")
dmu6 <- read.lp("DMU6.lp")

dmu1
```

DMUs(1,2,3,4,5,6)

```
## Model name:
##               u1     u2     v1     v2
## Maximize  14000   3500      0      0
## R1        14000   3500   -150   -200  <=  0
## R2        14000  21000   -400   -700  <=  0
## R3        42000  10500   -320  -1200  <=  0
## R4        28000  42000   -520  -2000  <=  0
## R5        19000  25000   -350  -1200  <=  0
## R6        14000  15000   -320   -700  <=  0
## R7            0      0    150    200   =  1
## Kind        Std    Std    Std    Std
## Type       Real   Real   Real   Real
## Upper       Inf    Inf    Inf    Inf
## Lower         0      0      0      0
```

```
solve(dmu1)
```

```
## [1] 0
```

```
get.objective(dmu1)
```

```
## [1] 1
```

```
get.variables(dmu1)
```

```
## [1] 7.142857e-05 0.000000e+00 5.172414e-03 1.120690e-03
```

```
solve(dmu2)
```

```
## [1] 0
```

```
get.objective(dmu2)
```

```
## [1] 1
```

```
get.variables(dmu2)
```

```
## [1] 0.000000e+00 4.761905e-05 1.376147e-03 6.422018e-04
```

```
solve(dmu3)
```

```
## [1] 0
```

```
get.objective(dmu3)
```

```
## [1] 1
```

```r
get.variables(dmu3)
```

```
## [1] 2.380952e-05 0.000000e+00 1.724138e-03 3.735632e-04
```

```r
solve(dmu4)
```

```
## [1] 0
```

```r
get.objective(dmu4)
```

```
## [1] 1
```

```r
get.variables(dmu4)
```

```
## [1] 0.000000e+00 2.380952e-05 6.880734e-04 3.211009e-04
```

```r
solve(dmu5)
```

```
## [1] 0
```

```r
get.objective(dmu5)
```

```
## [1] 0.9774987
```

```r
get.variables(dmu5)
```

```
## [1] 0.0000115123 0.0000303506 0.0010989011 0.0005128205
```

```r
solve(dmu6)
```

```
## [1] 0
```

```r
get.objective(dmu6)
```

```
## [1] 0.8674521
```

```r
get.variables(dmu6)
```

```
## [1] 1.620029e-05 4.270987e-05 1.546392e-03 7.216495e-04
```

We can see from the result above that DMUs 5 and 6 aren't efficient(i.e. the objectives aren't 1). To be specific, DMU5 is at 0.977 and DMU6 is at 0.867. But we can achieve maximum efficiency for DMUs 1,2,3,and 4. This happens when:
* For DMU1: weights of 7.142857e-05 and 0 for outputs; 5.172414e-03 and 1.120690e-03 for the two inputs
* For DMU2: weights of 0 and 4.761905e-05 for outputs; 1.376147e-03 and 6.422018e-04 for the two inputs
* For DMU3: weights of 2.380952e-05 and 0 for outputs; 1.724138e-03 and 3.735632e-04 for the inputs
* For DMU4: 0 and 2.380952e-05 for outputs; 6.880734e-04 and 3.211009e-04 for the inputs

```r
#install.packages("Benchmarking")
library(Benchmarking)
```

```
## Loading required package: ucminf

## Loading required package: quadprog
```

```r
x <- matrix(c(150,400,320,520,350,320,200,700,1200,2000,1200,700),ncol = 2)
y <- matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol = 2)
colnames(y) <- c("ReimbursedPat-Days","PrvtPaidPat-Days")
colnames(x) <- c("StaffHrsPerDay","SuppliesPerDay")
x
```

```
##      StaffHrsPerDay SuppliesPerDay
## [1,]            150            200
## [2,]            400            700
## [3,]            320           1200
## [4,]            520           2000
## [5,]            350           1200
## [6,]            320            700
```

```r
y
```

```
##      ReimbursedPat-Days PrvtPaidPat-Days
## [1,]              14000             3500
## [2,]              14000            21000
## [3,]              42000            10500
## [4,]              28000            42000
## [5,]              19000            25000
## [6,]              14000            15000
```

this is just a place holder

```r
eff.crs <- dea(x,y,RTS = "crs")
eff.crs
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675
```

```r
peers(eff.crs)
```

```
##      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
```

```r
lambda(eff.crs)
```

```
##              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
```

```r
eff.vrs <- dea(x,y,RTS = "vrs")
eff.vrs
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963
```

```r
peers(eff.vrs)
```

```
##      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
```

```r
lambda(eff.vrs)
```

```
##              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
```

```r
eff.fdh <- dea(x,y,RTS = "fdh")
eff.fdh
```

```
## [1] 1 1 1 1 1 1
```

```r
peers(eff.fdh)
```

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

```r
lambda(eff.fdh)
```

```
##      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
```

```r
eff.irs <- dea(x,y,RTS = "irs")
eff.irs
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963
```

```r
peers(eff.irs)
```

```
##      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
```

```r
lambda(eff.irs)
```

```
##             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
```

```r
eff.drs <- dea(x,y,RTS = "drs")
eff.drs
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675
```

```r
peers(eff.drs)
```

```
##      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
```

```r
lambda(eff.drs)
```

```
##             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
```

```r
eff.frh <- dea(x,y,RTS = "add")
eff.frh
```

```
## [1] 1 1 1 1 1 1
```

```r
peers(eff.frh)
```

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

```r
lambda(eff.frh)
```

```
##      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
```

Facility	CRS Efficiency	Peers	Lambda	VRS Efficiency	Peers	Lambda	FDH Efficiency	IRS Efficiency	Peers	Lambda	DRS Efficiency	Peers	Lambda	FRH Efficiency
Facility 1	1	1		1	1		1	1	1		1	1		1
Facility 2	1	2		1	2		1	1	2		1	2		1
Facility 3	1	3		1	3		1	1	3		1	3		1
Facility 4	1	4		1	4		1	1	4		1	4		1
Facility 5	0.9775	1,2,4	0.2,0.08, 0.538	1	5		1	1	5		0.9775	1,2,4	0.2,0.08, 0.538	1
Facility 6	0.8675	1, 2,4	0.34,0.39 ,0.13	0.8963	1,2,5	0.4,0.34, 0.26	1	0.8963	1,2,5	0.4,0.34, 0.26	0.8675	1, 2,4	0.34,0.39 ,0.13	1

- *Under CRS:* DMUs 1,2,3,and 4 are 100% efficient. DMU5 is 97.75% efficient, and DMU6 is only 86.75% efficient.
  - The peer units for DMU5 are 1,2,and 4, with relative weights(lambda) of 0.2, 0.08,and 0.538

  - The peer units for DMU6 are 1,2,and 4, with relative weights(lambda) of 0.34, 0.39,and 0.13

- *Under VRS:* DMUs 1,2,3,4,5 are 100% efficient. DMU6 is 89.63% efficient.

  - The peer units for DMU6 are 1,2,and 5, with relative weights(lambda) of 0.40, 0.34,and 0.26

- *Under FDH:* All DMUs are 100% efficient.

- *Under IRS:* DMUs 1,2,3,4,5 are 100% efficient. DMU6 is 89.63% efficient.

  - The peer units for DMU6 are 1,2,and 5, with relative weights(lambda) of 0.40, 0.34,and 0.26

- *Under DRS:* DMUs 1,2,3,and 4 are 100% efficient. DMU5 is 97.75% efficient, and DMU6 is only 86.75% efficient.

  - The peer units for DMU5 are 1,2,and 4, with relative weights(lambda) of 0.2, 0.08,and 0.538
  - The peer units for DMU6 are 1,2,and 4, with relative weights(lambda) of 0.34, 0.39,and 0.13

- *Under FRH:* All DMUs are 100% efficient.

Question 2:

The Research and Development Division of the Emax Corporation has developed three new products. A decision now needs to be made on which mix of these products should be produced. Management wants primary consideration given to three factors: total profit, stability in the workforce, and achieving an increase in the company's earnings next year from the $75 million achieved this year. In particular, using the units given in the following table, they want to Maximize Z = P - 6C - 3D, where:
* P = total (discounted) profit over the life of the new products,
* C = change (in either direction) in the current level of employment,
* D = decrease (if any) in next year's earnings from the current year's level.

Factor	P1	P2	P3	Goal	Units
Total Profit	20	15	25	Maximize	$ Millions
Facility 2	6	4	5	= 50	100s of employees
Facility 3	8	7	5	>= 75	$ Millions

Part 2.1: Define y1+ and y1-, respectively, as the amount over (if any) and the amount under (if any) the employment level goal. Define y2+ and y2- in the same way for the goal regarding earnings next year. Define x1, x2, and x3 as the production rates of Products 1, 2, and 3, respectively. With these definitions, use the goal programming technique to express y1+, y1-, y2+ and y2- algebraically in terms of x1, x2, and x3. Also express P in terms of x1, x2, and x3. 20x1 + 15x2 + 25x3 = P . . . . . . This is for the total profit
6x1 + 4x2 + 5x3 = 50 . . . . . . This is for workforce stability goal
8x1 + 7x2 + 5x3 >= 75 . . . . . This is for the earnings goal

Introducing auxiliary variables:
y1 = 6x1 + 4x2 + 5x3 - 50
y2 = 8x1 + 7x2 + 5x3 - 75

since: y1 = (y1P - y1N) and y2 = (y2P - y2N)

6x1 + 4x2 + 5x3 - (y1P - y1N) = 50
8x1 + 7x2 + 5x3 - (y2P - y2N) = 75
P = 20x1 + 15x2 + 25x3
x1,x2,x3,y1P,y1N,y2P,y2N >= 0 non-negativity constraint

Part 2.2: What is Management's objective function in terms of x1, x2, x3, y1P, y1N, y2P, y2N ? Maximize Z = 20x1 + 15x2 + 25x3 - 6 y1P - 6 y1N - 3 y2N

As employment goal is in both directions, it includes y1P and y1N; Earnings is only if we go under i.e. only y2N.

Part 2.3: Formulate and solve the linear programming model. What are your findings?

```
/* DMU1: Objective function */

max: 20 x1 + 15 x2 + 25 x3 - 6 y1P - 6 y1N - 3 y2N;

/* Constraints */

6x1 + 4x2 + 5x3 - (y1P - y1N) = 50;
8x1 + 7x2 + 5x3 - (y2P - y2N) = 75;
```

Reading the goal formulation from GP.lp file and assign it to gp

```r
gp <- read.lp("GP.lp")
gp
```

```
## Model name:
##              x1    x2    x3   y1P   y1N   y2N   y2P
## Maximize     20    15    25    -6    -6    -3     0
## R1            6     4     5    -1     1     0     0  =  50
## R2            8     7     5     0     0     1    -1  =  75
## Kind        Std   Std   Std   Std   Std   Std   Std
## Type       Real  Real  Real  Real  Real  Real  Real
## Upper       Inf   Inf   Inf   Inf   Inf   Inf   Inf
## Lower         0     0     0     0     0     0     0
```

```r
solve(gp)
```

```
## [1] 0
```

```r
# the objective value for this problem is:
get.objective(gp)
```

```
## [1] 225
```

```r
# the values of decision variables for the optimal solution are:
get.variables(gp)
```

```
## [1]  0  0 15 25  0  0  0
```

Z = 225
x1=0, x2=0, x3=15
y1P=25, y1N=0, y2N=0, y2P=0

The employment goal is exceeded by 25 employees and the resulting penalty from exceeding the goal is 225.

The total (discounted) profit over the life of the new products would be 25*15 = 375 million.