Assignment 6

SABIHA MHATARNAIK

11/21/2021

Question 1

Consider the following activity-on-arc project network, where the 12 arcs (arrows) represent the 12 activities (tasks) that must be performed to complete the project and the network displays the order in which the activities need to be performed. The number next to each arc (arrow) is the time required for the corresponding activity. Consider the problem of finding the longest path (the largest total time) through this network from start (node 1) to finish (node 9), since the longest path is the critical path.

Solution:

Objective Function

Maximize: $5X_{12}$ + $3X_{13}$ + $4X_{24}$ + $2X_{25}$ + $3X_{35}$ + X_{46} + $4X_{47}$ + $6X_{57}$ + $2X_{58}$ + $5X_{69}$ + $4X_{79}$ + $7X_{89}$
Here ,X_{ij} (i=Starting Node ,j=Ending node)

Starting Node

$5X_{12}$ + $3X_{13}$ =1

Intermediate Node

$5X_{12}$ - $2X_{25}$ - $4X_{24}$ =0

$3X_{13}$ - $3X_{35}$ = 0

$4X_{24}$ - X_{46} - $4X_{47}$ = 0

$3X_{35}$ - $2X_{25}$ - $2X_{58}$ - $6X_{57}$ = 0

X_{46} - $5X_{69}$ = 0

$6X_{57}$ - $4X_{47}$ - $4X_{24}$ =0

$2X_{58}$ - $7X_{89}$ = 0

Ending Node

$7X_{89}$ - $4X_{79}$ - $5X_{69}$ = 1

```
library(lpSolveAPI)
lprec <- make.lp(0,12)
lp.control(lprec, sense="max")
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
```

```
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
```

```
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
time <- c(5, 3, 4, 2, 3, 1, 4, 6, 2, 5, 4, 7) # objective function
set.objfn(lprec, 1*time)
set.type(lprec, 1:12, "binary")
add.constraint(lprec,c(1,1),"=",1,indices = c(1,2)) # starting node
add.constraint(lprec,c(1,-1,-1),"=",0,indices = c(1,3,4)) # intermediate node
add.constraint(lprec,c(1,-1),"=",0,indices = c(2,5))
add.constraint(lprec,c(1,-1,-1),"=",0,indices = c(3,6,7))
add.constraint(lprec,c(1,1,-1,-1),"=",0,indices = c(4,5,8,9))
add.constraint(lprec,c(1, -1),"=",0,indices = c(6,10))
add.constraint(lprec,c(1,1,-1),"=",0,indices = c(7,8,11))
add.constraint(lprec,c(1,-1),"=",0,indices = c(9,12))
add.constraint(lprec,c(1,1,1),"=",1,indices = c(10,11,12)) # Ending node
solve(lprec)
```

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

```
get.objective(lprec)
```

```
## [1] 17
```

```
get.variables(lprec)
```

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

```
get.constraints(lprec)
```

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

```
arc <- c("X12", "X13", "X24", "X25", "X35", "X46", "X47", "X57", "X58", "X69", "X79", "X89")
var<-get.variables(lprec)
output<-data.frame(arc,var)
output
```

```
##    arc var
## 1  X12   1
## 2  X13   0
## 3  X24   0
## 4  X25   1
## 5  X35   0
## 6  X46   0
## 7  X47   0
## 8  X57   1
```

```
## 9  X58   0
## 10 X69   0
## 11 X79   1
## 12 X89   0
```

Also ,we can read the .lp file and solve the problem

```
x<- read.lp("BIP.lp")
x
```

```
## Model name:
##   a linear program with 12 decision variables and 9 constraints
```

Solve Model

```
solve(x)
```

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

```
get.objective(x)
```

```
## [1] 17
```

```
get.variables(x)
```

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

```
get.constraints(x)
```

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

Question 2

Selecting an Investment Portfolio An investment manager wants to determine an opti- mal portfolio for a wealthy client. The fund has $2.5 million to invest, and its objective is to maximize total dollar return from both growth and dividends over the course of the coming year. The client has researched eight high-tech companies and wants the portfolio to consist of shares in these firms only. Three of the firms (S1 − S3) are primarily software companies, three (H1–H3) are primarily hardware companies, and two (C1–C2) are internet consulting companies. The client has stipulated that no more than 40 percent of the investment be allocated to any one of these three sectors. To assure diversification, at least $100,000 must be invested in each of the eight stocks. Moreover, the number of shares invested in any stock must be a multiple of 1000.

The table gives estimates from the investment company's database relating to these stocks. These estimates include the price per share, the projected annual growth rate in the share price, and the anticipated annual dividend payment per share.

1) Determine the maximum return on the portfolio. What is the optimal number of shares to buy for each of the stocks? What is the corresponding dollar amount invested in each stock?

Solution:

To determine the maximum return on the portfolio the objective function includes the price per share, the projected annual growth in the share price & the anticipated annual dividend payment per share .This can be expressed as follows :

Returns =(Price per share) * (Growth Rate) + (Dividend payment per share)

Objective Function

Maximize, Z = $4XS_1 + 6.5XS_2 + 5.9XS_3 + 5.4XH_1 + 5.15XH_2 + 10XH_3 + 8.4XC_1 + 6.25XC_2$

Investment constraints,

40 XS_1 + 50 XS_2 + 80 XS_3 + 60 XH_1 + 45 XH_2 + 60 XH_3 + 30 XC_1 + 25 XC_2 <= 2500000

Number of shares that are invested in stock should be a multiple of 1000.

1000 X_{SI} >= 0 (I = 1,2,3)
1000 X_{HI} >= 0 (I = 1,2,3)
1000 X_{CI} >= 0 (I = 1,2)

Also, at least $100,000 must be invested in each of the eight stocks.

40 X_{S1} >= 100000;

50 X_{S2} >= 100000;

80 X_{S3} >= 100000;

60 X_{H1} >= 100000;

45 X_{H2} >= 100000;

60 X_{H3} >= 100000;

30 X_{C1} >= 100000;

25 X_{C2} >= 100000

Also, no more than 40% of the investment should be allocated to these 3 sectors

40 X_{S1} + 50 X_{S2} + 80 X_{S3} <= 1000000

60 X_{H1} + 45 X_{H2}+ 60 X_{H3} <= 1000000

30 X_{C1} + 25 X_{C2} <= 1000000

where X_{SI}, X_{HI}, X_{CI} >= 0 are integers.

```
library(lpSolveAPI)
lprec<-make.lp(0,8)
lp.control(lprec,sense="max")
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
```

```
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
```

```
## [1] "neutral"
```

```
set.objfn(lprec,c(4,6.5,5.9,5.4,5.15,10,8.4,6.25))
set.type(lprec,c(1:8),type = "integer")
add.constraint(lprec,c(40,50,80,60,45,60,30,25),"<=",2500000,indices = c(1:8))
add.constraint(lprec,1000,">=",0,indices = 1)
add.constraint(lprec,1000,">=",0,indices = 2)
add.constraint(lprec,1000,">=",0,indices = 3)
add.constraint(lprec,1000,">=",0,indices = 4)
add.constraint(lprec,1000,">=",0,indices = 5)
add.constraint(lprec,1000,">=",0,indices = 6)
add.constraint(lprec,1000,">=",0,indices = 7)
add.constraint(lprec,1000,">=",0,indices = 8)
add.constraint(lprec,40,">=",100000,indices = 1)
add.constraint(lprec,50,">=",100000,indices = 2)
add.constraint(lprec,80,">=",100000,indices = 3)
add.constraint(lprec,60,">=",100000,indices = 4)
add.constraint(lprec,45,">=",100000,indices = 5)
add.constraint(lprec,60,">=",100000,indices = 6)
add.constraint(lprec,30,">=",100000,indices = 7)
add.constraint(lprec,25,">=",100000,indices = 8)
add.constraint(lprec,c(40,50,80),"<=",1000000,indices = c(1,2,3))
add.constraint(lprec,c(60,45,60),"<=",1000000,indices = c(4,5,6))
add.constraint(lprec,c(30,25),"<=",1000000,indices = c(7,8))
solve(lprec)
```

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

```
get.objective(lprec)
```

```
## [1] 487145.2
```

```
get.variables(lprec)
```

```
## [1]  2500  6000  1250  1667  2223 13332 30000  4000
```

```
get.constraints(lprec)
```

```
##  [1]  2499975  2500000  6000000  1250000  1667000  2223000 13332000 30000000
##  [9]  4000000   100000   300000   100000   100020   100035   799920   900000
## [17]   100000   500000   999975  1000000
```

Using lpsolve with integer restriction we get the objective function, maximum returns as 487145.2 and number of stocks are S1= 2500, S2= 6000, S3= 1250, H1= 1667, H2= 2223, H3= 13332, C1= 30000, C2= 4000.

The amount that is invested in each stock S1= 100000, S2= 300000, S3= 100000, H1= 100020, H2= 100035, H3= 799920, C1= 900000, C2= 100000.

2) Compare the solution in which there is no integer restriction on the number of shares invested. By how much (in percentage terms) do the integer restrictions alter the value of the optimal objective function? By how much (in percentage terms) do they alter the optimal investment quantities?

Solution:

```
library(lpSolveAPI)
lprec<-make.lp(0,8)
lp.control(lprec,sense="max")
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
```

```
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
set.objfn(lprec,c(4,6.5,5.9,5.4,5.15,10,8.4,6.25))
add.constraint(lprec,c(40,50,80,60,45,60,30,25),"<=",2500000,indices = c(1:8))
add.constraint(lprec,1000,">=",0,indices = 1)
add.constraint(lprec,1000,">=",0,indices = 2)
add.constraint(lprec,1000,">=",0,indices = 3)
add.constraint(lprec,1000,">=",0,indices = 4)
add.constraint(lprec,1000,">=",0,indices = 5)
add.constraint(lprec,1000,">=",0,indices = 6)
add.constraint(lprec,1000,">=",0,indices = 7)
add.constraint(lprec,1000,">=",0,indices = 8)
add.constraint(lprec,40,">=",100000,indices = 1)
add.constraint(lprec,50,">=",100000,indices = 2)
add.constraint(lprec,80,">=",100000,indices = 3)
add.constraint(lprec,60,">=",100000,indices = 4)
add.constraint(lprec,45,">=",100000,indices = 5)
add.constraint(lprec,60,">=",100000,indices = 6)
add.constraint(lprec,30,">=",100000,indices = 7)
add.constraint(lprec,25,">=",100000,indices = 8)
add.constraint(lprec,c(40,50,80),"<=",1000000,indices = c(1,2,3))
add.constraint(lprec,c(60,45,60),"<=",1000000,indices = c(4,5,6))
add.constraint(lprec,c(30,25),"<=",1000000,indices = c(7,8))
solve(lprec)
```

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

```r
get.objective(lprec)
```

```
## [1] 487152.8
```

```r
get.variables(lprec)
```

```
## [1]  2500.000  6000.000  1250.000  1666.667  2222.222 13333.333 30000.000
## [8]  4000.000
```

```r
get.constraints(lprec)
```

```
##  [1]  2500000  2500000  6000000  1250000  1666667  2222222 13333333 30000000
##  [9]  4000000   100000   300000   100000   100000   100000   800000   900000
## [17]   100000   500000  1000000  1000000
```

Using lpsolve without integer restriction we get the objective function, maximum returns as 487152.8 and number of stocks are S1= 2500.0, S2= 6000.0, S3= 1250.0, H1= 1666.667, H2= 2222.222, H3= 13333.333, C1= 30000.0, C2= 4000.0

The amount that is invested in each stock S1= 100000, S2= 300000, S3= 100000, H1= 100000, H2= 100000, H3= 800000, C1= 900000, C2= 100000.

The Percentage difference in objective functions with and without integer restriction is 0.00156