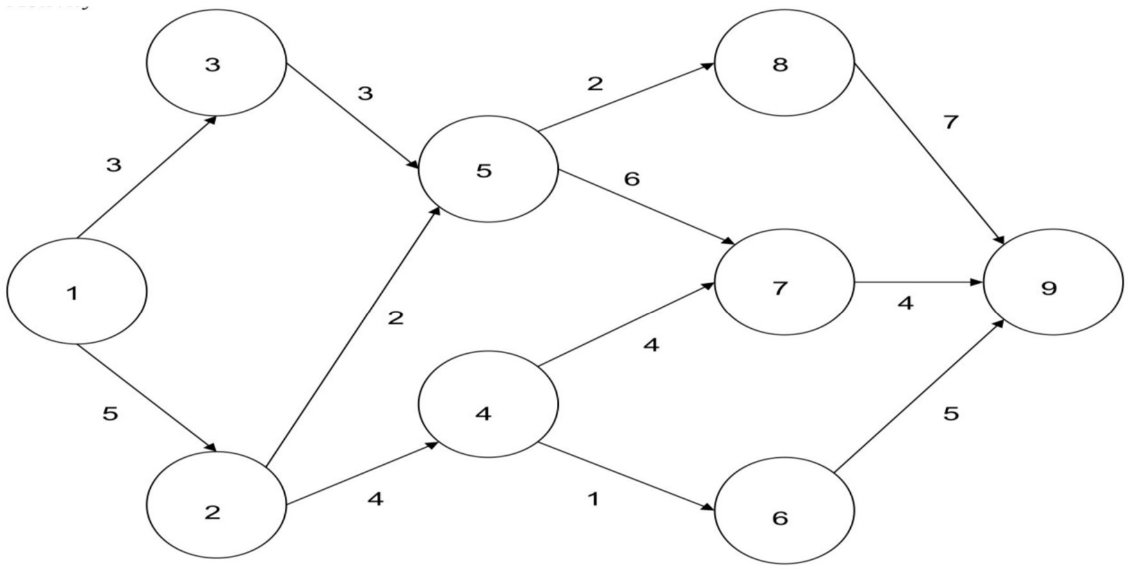
Assignment 6

Niharika Pillanagoyala

Question 1:

Formulate and solve the binary integer programming (BIP) model for this problem using library lpsolve or equivalent in R.



Solution:

Zmax is the objective function, and the longest path is the critical path;
Zmax= 3X13 + 5X12 + 3X35 + 2X25 + 2X58 + 4X24 + 6X57 + 4X47 + 1X46 + 7X89 + 4X79 + 5X69
where Xij(i=starting node, j= ending node)

Starting node:
Node 1 :3X13 + 5X12 = 1

Ending node:
Node 9: 7X89 + 4X79 + 5X69 = 1

Intermediate nodes:
Node 2: 5X12 = 2X25 + 4X24
Node 3: 3X13 - 3X35 = 0

Node 4: 4X24 - 1X46 = 4X47
Node 5: 3X35 + 2X25 = 2X58 + 6X57
Node 6: 1X46 = 5X69
Node 7:6X57 + 4X47 = 4X79
Node 8: 2X58 = 7X89
Where Xij are binary

The longest path is the critical path which is between the nodes (1-2-5-7-9)

So, the arcs between the objective function is X12-X25-X57-X79

R Program:

Loading libraries

```
library(lpSolveAPI)
lp <- make.lp(0,12)
lp.control(lp, 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"
```

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

```r
add.constraint(lp,c(1,-1),"=",0,indices = c(9,12))
# End node
add.constraint(lp,c(1,1,1),"=",1,indices = c(10,11,12))
solve(lp)

## [1] 0

get.objective(lp)

## [1] 17

get.variables(lp)

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

get.constraints(lp)

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

arc <- c("x12", "x13", "x24", "x25", "x35", "x46", "x47", "x57", "x58", "x69"
, "x79", "x89")
variables<-get.variables(lp)
output<-data.frame(arc,variables)
output

##    arc variables
## 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
```

Question 2:

Formulating the lp function.

2a) Solution

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?

```r
lps<-make.lp(0,8)
lp.control(lps,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"
```

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

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

```
get.objective(lps)
```

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

```
get.variables(lps)

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

get.constraints(lps)

##  [1]  2499975  2500000  6000000  1250000  1667000  2223000 13332000 300000
00
##  [9]  4000000   100000   300000   100000   100020   100035   799920   9000
00
## [17]   100000   500000   999975  1000000
```

According to the problem returns can be given by
Returns = (Price per share) * (Growth rate of share) + (Dividend per share)
Hence the objective function is
Zmax = 4XS1 + 6.5XS2 + 5.9XS3 + 5.4XH1 + 5.15XH2 + 10XH3 + 8.4XC1 + 6.25XC2

Constraints:

Investment constraint:
40XS1 + 50XS2 + 80XS3 + 60XH1 + 45XH2 + 60XH3 + 30XC1 + 25XC2 <= 2500000
The stock must be a multiple of 1000
1000XS1 >= 0; 1000XS2 >= 0; 1000XS3 >= 0
1000XH1 >= 0; 1000XH2 >= 0; 1000XH3 >= 0
1000XC1 >= 0; 1000XC2 >= 0

Maximum amount invested in 1 sector = 2.5 million * 40% = 1 million
No more than 40% should be assigned to the 3 sectors,

40XS1 + 50XS2 + 80XS3 <= 1000000
60XH1 + 45XH2 + 60XH3 <= 10000000

30XC1 + 25XC2 <= 1000000

Where XS1, XH1, XC1 >= 0

Minimum investment in each stock = .1 million
Least $100,000 must be invested in 8 stocks
40XS1 >= 100000
50XS2 >= 100000
80XS3 >= 100000
60XH1 >= 100000
45XH2 >= 100000
60XH3 >= 100000
30XC1 >= 100000
25XC2 >= 100000
Where XSJ, XHJ, XCJ >= 0 are integers.

Optimal number of shares to buy each of the stock

S1 = 100000/ 40 = 2500
S2= 100000/ 50 = 2000
S3= 300000/ 80 = 3750
H1= 100000/ 60 = 1666.67
H2 = 100000/ 45 = 2222.22
H3= 800000/ 60 = 13333.33
C1 = 900000/ 30 = 30000
C2 = 100000/ 25 = 4000

The amount invested in each stock

40% investment in sector 3 : C1= 900000, C2= 100000;
40% investment in sector 2 (H1 , H2, H3) : H1= 100000, H2= 100000, H3= 800000;
Balance in sector 1 (S1, S2, S3) : S1= 100000, S2= 300000, S3= 100000;

Formulating lp problem without integer restrictions

```
lps1<-make.lp(0,8)
lp.control(lps1,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"
```

```
#Formulating without integer
set.objfn(lps1,c(4,6.5,5.9,5.4,5.15,10,8.4,6.25))
add.constraint(lps1,c(40,50,80,60,45,60,30,25),"<=",2500000,indices = c(1:8))
add.constraint(lps1,1000,">=",0,indices = 1)
add.constraint(lps1,1000,">=",0,indices = 2)
add.constraint(lps1,1000,">=",0,indices = 3)
add.constraint(lps1,1000,">=",0,indices = 4)
add.constraint(lps1,1000,">=",0,indices = 5)
add.constraint(lps1,1000,">=",0,indices = 6)
```

```
add.constraint(lps1,1000,">=",0,indices = 7)
add.constraint(lps1,1000,">=",0,indices = 8)
add.constraint(lps1,40,">=",100000,indices = 1)
add.constraint(lps1,50,">=",100000,indices = 2)
add.constraint(lps1,80,">=",100000,indices = 3)
add.constraint(lps1,60,">=",100000,indices = 4)
add.constraint(lps1,45,">=",100000,indices = 5)
add.constraint(lps1,60,">=",100000,indices = 6)
add.constraint(lps1,30,">=",100000,indices = 7)
add.constraint(lps1,25,">=",100000,indices = 8)
add.constraint(lps1,c(40,50,80),"<=",1000000,indices = c(1,2,3))
add.constraint(lps1,c(60,45,60),"<=",1000000,indices = c(4,5,6))
add.constraint(lps1,c(30,25),"<=",1000000,indices = c(7,8))
solve(lps1)

## [1] 0

get.objective(lps1)

## [1] 487152.8

get.variables(lps1)

## [1]  2500.000  6000.000  1250.000  1666.667  2222.222 13333.333

[7] 30000.000  4000.000

get.constraints(lps1)
```

```
[1]  2500000  2500000  6000000  1250000  1666667  2222222 13333333

 [8] 30000000  4000000   100000   300000   100000   100000   100000

[15]   800000   900000   100000   500000  1000000  1000000
```

2b) Solution

2b) 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?

The number of stocks are:
S1 = 100000/ 40 = 2500
S2= 100000/ 50 = 2000
S3= 300000/ 80 = 3750
H1= 100000/ 60 = 1666.67

H2 = 100000/ 45 = 2222.22
H3= 800000/ 60 = 13333.33
C1 = 900000/ 30 = 30000
C2 = 100000/ 25 = 4000

The amount invested in each stocks:

40% investment in sector 3 : C1= 900000, C2= 100000;
40% investment in sector 2 (H1 , H2, H3) : H1= 100000, H2= 100000, H3= 800000;
Balance in sector 1 (S1, S2, S3) : S1= 100000, S2= 300000, S3= 100000;

Considering lp problem with integer restrictions and without integer restrictions there is a difference of $7.6(487152.8-487145.2).

The value of the objective function differs by 0.00156%.

The investment quantities are altered as follows: S1 = S2 = 0, S3 = 0%, H1 = 0.03996% increased, H2 = 0.03501% decreased, H3 = 75. 01% increased and C1 = C2 = 0