Assignment 3

Prajwal C N

10/8/2021

Weiglet Primal Solution

Q1. Solve the problem using lpSolve or any other equivalent library in R.

```
#Weiglet Primal Solution
library(lpSolveAPI)

lprec<-make.lp(0,9)
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"
```

```
# Objective function with 9 decision variables
set.objfn(lprec,rep(c(420,360,300),3))

# Total there are 11 Constraints
add.constraint(lprec,c(1,1,1),"<=",750,indices = c(1,2,3))
add.constraint(lprec,c(1,1,1),"<=",900,indices = c(4,5,6))
add.constraint(lprec,c(1,1,1),"<=",450,indices = c(7,8,9))
add.constraint(lprec,c(20,15,12),"<=",13000,indices = c(1,2,3))
add.constraint(lprec,c(20,15,12),"<=",12000,indices = c(4,5,6))
add.constraint(lprec,c(20,15,12),"<=",5000,indices = c(7,8,9))
add.constraint(lprec,c(1,1,1),"<=",900,indices = c(1,4,7))
add.constraint(lprec,c(1,1,1),"<=",1200,indices = c(2,5,8))
add.constraint(lprec,c(1,1,1),"<=",750,indices = c(3,6,9))
add.constraint(lprec,c(rep(c(900,-750),each=3)),"=",0,indices = c(1,2,3,4,5,6))
add.constraint(lprec,c(rep(c(450,-750),each=3)),"=",0,indices = c(1,2,3,7,8,9))
```

Solve LP problem

```
solve(lprec)
```

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

Output of above solve is 0,So an optimal solution was obtained

Now we can get the value of the objective function, and the variables.

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

```
## [1] 696000
```

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

```
##  [1]  6.944444e+02  8.333333e+02  4.166667e+02  1.300000e+04  1.200000e+04
##  [6]  5.000000e+03  5.166667e+02  8.444444e+02  5.833333e+02 -2.037268e-10
## [11]  0.000000e+00
```

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

```
## [1] 516.6667 177.7778   0.0000   0.0000 666.6667 166.6667   0.0000   0.0000
## [9] 416.6667
```

From the above solution, we can infer the following : Plant 1 : 516.67 of Large Products and 177.78 of Medium Products. Plant 2 : 666.67 of Medium Products and 166.67 of Small products. Plant 3 : 416.67 of Small Products

Q2. Identify the shadow prices,dual solution and reduced costs.

```
# Reduced Cost
get.sensitivity.obj(lprec)
```

```
## $objfrom
## [1]  3.60e+02  3.45e+02 -1.00e+30 -1.00e+30  3.45e+02  2.52e+02 -1.00e+30
## [8] -1.00e+30  2.04e+02
##
## $objtill
## [1] 4.60e+02 4.20e+02 3.24e+02 4.60e+02 4.20e+02 3.24e+02 7.80e+02 4.80e+02
## [9] 1.00e+30
```

Reduced cost mentioned in $objfrom and $objtill.

```
# Shadow prices
get.sensitivity.rhs(lprec)
```

```
## $duals
##  [1]    0.00    0.00    0.00   12.00   20.00   60.00    0.00    0.00    0.00
## [10]   -0.08    0.56    0.00    0.00  -24.00  -40.00    0.00    0.00 -360.00
## [19] -120.00    0.00
##
## $dualsfrom
##  [1] -1.000000e+30 -1.000000e+30 -1.000000e+30  1.122222e+04  1.150000e+04
##  [6]  4.800000e+03 -1.000000e+30 -1.000000e+30 -1.000000e+30 -2.500000e+04
## [11] -1.250000e+04 -1.000000e+30 -1.000000e+30 -2.222222e+02 -1.000000e+02
## [16] -1.000000e+30 -1.000000e+30 -2.000000e+01 -4.444444e+01 -1.000000e+30
##
## $dualstill
##  [1] 1.000000e+30 1.000000e+30 1.000000e+30 1.388889e+04 1.250000e+04
##  [6] 5.181818e+03 1.000000e+30 1.000000e+30 1.000000e+30 2.500000e+04
## [11] 1.250000e+04 1.000000e+30 1.000000e+30 1.111111e+02 1.000000e+02
## [16] 1.000000e+30 1.000000e+30 2.500000e+01 6.666667e+01 1.000000e+30
```

Shadow prices are mentioned in $duals, and the ranges for shadow price calculations present under $dualsfrom and $dualstill.

Q3. Further,identify the sensitivity of the above prices and costs.That is,specify the range of shadow prices and reduced cost within which the optimal solution will not change

```
s1 <- data.frame(get.sensitivity.rhs(lprec)$duals[1:11],get.sensitivity.rhs(lprec)$dualsfrom[1:11],get.sensitivit
y.rhs(lprec)$dualstill[1:11])
names(s1)<-c("price","lower","upper")
s2 <- data.frame(get.sensitivity.rhs(lprec)$duals[12:20],get.sensitivity.rhs(lprec)$dualsfrom[12:20],get.sensitiv
ity.rhs(lprec)$dualstill[12:20])
names(s2)<-c("cost","lower","upper")

s1
```

```
##    price         lower        upper
## 1   0.00 -1.000000e+30 1.000000e+30
## 2   0.00 -1.000000e+30 1.000000e+30
## 3   0.00 -1.000000e+30 1.000000e+30
## 4  12.00  1.122222e+04 1.388889e+04
## 5  20.00  1.150000e+04 1.250000e+04
## 6  60.00  4.800000e+03 5.181818e+03
## 7   0.00 -1.000000e+30 1.000000e+30
## 8   0.00 -1.000000e+30 1.000000e+30
## 9   0.00 -1.000000e+30 1.000000e+30
## 10 -0.08 -2.500000e+04 2.500000e+04
## 11  0.56 -1.250000e+04 1.250000e+04
```

```
s2
```

```
##   cost         lower        upper
## 1    0 -1.000000e+30 1.000000e+30
## 2    0 -1.000000e+30 1.000000e+30
## 3  -24 -2.222222e+02 1.111111e+02
## 4  -40 -1.000000e+02 1.000000e+02
## 5    0 -1.000000e+30 1.000000e+30
## 6    0 -1.000000e+30 1.000000e+30
## 7 -360 -2.000000e+01 2.500000e+01
## 8 -120 -4.444444e+01 6.666667e+01
## 9    0 -1.000000e+30 1.000000e+30
```

sensitivity of the above prices and costs are mentioned the range of shadow prices comprising of $dualsfrom and $dualstill.

The above is the range of shadow prices within which the optimal solution will not change.

Q4. Formulation of the dual of the above problem

Duality theory

```
#Duality theory
library(lpSolveAPI)
lprec<-make.lp(0,11)
lp.control(lprec,sense='min')
```

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

```
# Objective function
set.objfn(lprec,c(750,900,450,13000,12000,5000,900,1200,750,0,0))

add.constraint(lprec,c(1,20,1,900,450),">=",420,indices = c(1,4,9,10,11))
add.constraint(lprec,c(1,15,1,900,450),">=",360,indices = c(1,4,8,10,11))
add.constraint(lprec,c(1,12,1,900,450),">=",300,indices = c(1,4,7,10,11))
add.constraint(lprec,c(1,20,1,-750),">=",420,indices = c(2,5,9,10))
add.constraint(lprec,c(1,15,1,-750),">=",360,indices = c(2,5,8,10))
add.constraint(lprec,c(1,12,1,-750),">=",300,indices = c(2,5,7,10))
add.constraint(lprec,c(1,20,1,-750),">=",420,indices = c(3,6,9,11))
add.constraint(lprec,c(1,15,1,-750),">=",360,indices = c(3,6,8,11))
add.constraint(lprec,c(1,12,1,-750),">=",300,indices = c(3,6,7,11))

set.bounds(lprec,lower = c(-Inf,-Inf),columns = 10:11)
```

Solve LP problem

```
solve(lprec)
```

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

Output of above solve is 0,So an optimal solution was obtained

Now we can get the value of the objective function, and the variables.

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

```
## [1] 696000
```

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

```
## [1] 420 360 324 460 360 300 780 480 300
```

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

```
##  [1]  0.00  0.00  0.00 12.00 20.00 60.00  0.00  0.00  0.00 -0.08  0.56
```

```
# reduced costs
get.sensitivity.obj(lprec)
```

```
## $objfrom
##  [1]    694.4444    833.3333    416.6667  11222.2222  11050.0000   4800.0000
##  [7]    583.3333    844.4444    516.6667 -25000.0000 -12500.0000
##
## $objtill
##  [1] 1.000000e+30 1.000000e+30 1.000000e+30 1.388889e+04 1.250000e+04
##  [6] 5.345455e+03 1.000000e+30 1.000000e+30 1.000000e+30 1.000000e+30
## [11] 2.375000e+04
```

Reduced cost mentioned in $objfrom and $objtill.

The above is the range of Reduced cost within which the optimal solution will not change.

```
# shadow prices
get.sensitivity.rhs(lprec)
```

```
## $duals
##  [1] 516.66667 177.77778   0.00000   0.00000 666.66667 166.66667   0.00000
##  [8]   0.00000 416.66667  55.55556  66.66667  33.33333   0.00000   0.00000
## [15]   0.00000 316.66667 355.55556 233.33333   0.00000   0.00000
##
## $dualsfrom
##  [1]  3.600000e+02  3.450000e+02 -1.000000e+30 -1.000000e+30  3.450000e+02
##  [6]  2.880000e+02 -1.000000e+30 -1.000000e+30  2.040000e+02 -1.000000e+30
## [11] -5.915411e+13 -1.000000e+30 -1.000000e+30 -1.000000e+30 -1.000000e+30
## [16] -2.400000e+01 -1.500000e+01 -4.000000e+01 -1.000000e+30 -1.000000e+30
##
## $dualstill
##  [1] 4.60e+02 4.20e+02 1.00e+30 1.00e+30 3.75e+02 3.24e+02 1.00e+30 1.00e+30
##  [9] 1.00e+30 2.52e+02 6.00e+01 4.80e+02 1.00e+30 1.00e+30 1.00e+30 1.20e+01
## [17] 1.50e+01 6.00e+01 1.00e+30 1.00e+30
```

Shadow prices are mentioned in $duals, and the ranges for shadow price calculations present under $dualsfrom and $dualstill.

The above is the range of shadow prices within which the optimal solution will not change.

The Formulation of dual solution agrees with primal solution.