Assignment 3

Noorah

10/10/2021

The WeigeltCorporation has three branch plants with excess production capacity. Fortunately, the corporation has a new product ready to begin production, and all three plants have this capability, so some of the excess capacity can be used in this way. This product can be made in three sizes–large, medium, and small–that yield a net unit profit of $420, $360, and $300, respectively. Plants 1, 2, and 3 have the excess capacity to produce 750, 900, and 450 units per day of this product, respectively, regardless of the size or combination of sizes involved.The amount of available in-process storage space also imposes a limitation on the production rates of the new product. Plants 1, 2, and 3 have 13,000, 12,000, and 5,000 square feet, respectively, of in-process storage space available for a day's production of this product. Each unit of the large, medium, and small sizes produced per day requires 20, 15, and 12 square feet, respectively.Sales forecasts indicate that if available, 900, 1,200, and 750 units of the large, medium, and small sizes, respectively, would be sold per day.At each plant, some employees will need to be laid off unless most of the plant's excess production capacity can be used to produce the new product. To avoid layoffs if possible, management has decided that the plants should use the same percentage of their excess capacity to produce the new product.Management wishes to know how much of each of the sizes should be produced by each of the plants to maximize profit.

```
### Adding packege
library(lpSolveAPI)
```

```
### creating lp with o constraints and 9 decision variables
lprec <-make.lp(0,9)
lprec
```

```
## Model name:
##   a linear program with 9 decision variables and 0 constraints
```

```
### Using Max function to maximize the profit
set.objfn(lprec, c(420,360,300,420,360,300,420,360,300))
lp.control(lprec,sense="max")
```

```
## $anti.degen
## [1] "none"
##
## $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"
```

```
###Plant products production by day
add.constraint(lprec,c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=",750)
add.constraint(lprec,c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=",900)
add.constraint(lprec,c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=",450)

###Plant storage for products by day
add.constraint(lprec,c(20, 15, 12, 0, 0, 0, 0, 0, 0), "<=",13000)
add.constraint(lprec,c(0, 0, 0, 20, 15, 12, 0, 0, 0), "<=",12000)
add.constraint(lprec,c(0, 0, 0, 0, 0, 0, 20, 15, 12), "<=",5000)

###Sales for products by day:
add.constraint(lprec,c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=",900)
add.constraint(lprec,c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=",1200)
add.constraint(lprec,c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=",750)

###All plants should have the same percentage capacity to produce new products
add.constraint(lprec, c(6, 6, 6, -5, -5, -5, 0, 0, 0), "=",0)
add.constraint(lprec, c(3, 3, 3, 0, 0, 0, -5, -5, -5), "=",0)

RN <- c("CCon1", "CCon2", "CCon3", "SCon1", "SCon2", "SCon3", "SaCon1", "SaCon2", "Sa
Con3", "%C1", "%C2")
CN <- c("P1L", "P1M", "P1S", "P2L", "P2M", "P2S", "P3L", "P3M", "P3S")
dimnames(lprec) <- list(RN, CN)
lprec
```

```
## Model name:
##   a linear program with 9 decision variables and 11 constraints
```

```
write.lp(lprec, filename = "Assign2Q3", type = "lp")
```

```
### 1.Solve the problem using lpsolve, or any other equivalent library in R.
solve(lprec)
```

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

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

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

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

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

The profit is = $696000

```
### 2.Identify the shadow prices, dual solution, and reduced costs

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

```
## $duals
##  [1]    0    0    0   12   20   60    0    0    0  -12   84    0    0  -24  -40
## [16]    0    0 -360 -120    0
##
## $dualsfrom
##  [1] -1.000000e+30 -1.000000e+30 -1.000000e+30  1.041667e+04  1.000000e+04
##  [6]  4.800000e+03 -1.000000e+30 -1.000000e+30 -1.000000e+30 -3.333333e+02
## [11] -8.333333e+01 -1.000000e+30 -1.000000e+30 -8.611111e+02 -1.000000e+02
## [16] -1.000000e+30 -1.000000e+30 -5.000000e+01 -1.333333e+02 -1.000000e+30
##
## $dualstill
##  [1] 1.000000e+30 1.000000e+30 1.000000e+30 1.388889e+04 1.250000e+04
##  [6] 5.400000e+03 1.000000e+30 1.000000e+30 1.000000e+30 1.666667e+02
## [11] 1.666667e+02 1.000000e+30 1.000000e+30 1.111111e+02 2.500000e+02
## [16] 1.000000e+30 1.000000e+30 2.500000e+01 6.666667e+01 1.000000e+30
```

We can find Shadow prices under $dual and also calculations of shadow prices $dualsfrom and $dualstill

```
#Dual solution
get.dual.solution(lprec)
```

```
##  [1]    1    0    0    0   12   20   60    0    0    0  -12   84    0    0  -24
## [16]  -40    0    0 -360 -120    0
```

```
#Reduced costs
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
```

```
### 3.Identify the sensitivity of the above prices and costs.That is, specify the ran
ge of shadow prices and reduced cost within which the optimal solution will not chang
e.


##Shadow prices optimal solution will not change
cbind( get.sensitivity.rhs(lprec)$duals[1:21],lowerRange=get.sensitivity.rhs(lprec)$d
ualsfrom[1:21],upperRange=get.sensitivity.rhs(lprec)$dualstill[1:21])
```

```
##                lowerRange   upperRange
##  [1,]    0 -1.000000e+30 1.000000e+30
##  [2,]    0 -1.000000e+30 1.000000e+30
##  [3,]    0 -1.000000e+30 1.000000e+30
##  [4,]   12  1.041667e+04 1.388889e+04
##  [5,]   20  1.000000e+04 1.250000e+04
##  [6,]   60  4.800000e+03 5.400000e+03
##  [7,]    0 -1.000000e+30 1.000000e+30
##  [8,]    0 -1.000000e+30 1.000000e+30
##  [9,]    0 -1.000000e+30 1.000000e+30
## [10,]  -12 -3.333333e+02 1.666667e+02
## [11,]   84 -8.333333e+01 1.666667e+02
## [12,]    0 -1.000000e+30 1.000000e+30
## [13,]    0 -1.000000e+30 1.000000e+30
## [14,]  -24 -8.611111e+02 1.111111e+02
## [15,]  -40 -1.000000e+02 2.500000e+02
## [16,]    0 -1.000000e+30 1.000000e+30
## [17,]    0 -1.000000e+30 1.000000e+30
## [18,] -360 -5.000000e+01 2.500000e+01
## [19,] -120 -1.333333e+02 6.666667e+01
## [20,]    0 -1.000000e+30 1.000000e+30
## [21,]   NA            NA           NA
```

```
###Reduced costs optimal sloution will not change
cbind(get.sensitivity.obj(lprec)$duals[1:9],lowerRange=get.sensitivity.obj(lprec)$obj
from[1:9],upperRange=get.sensitivity.obj(lprec)$objtill[1:9])
```

```
##       lowerRange upperRange
##  [1,]   3.60e+02   4.60e+02
##  [2,]   3.45e+02   4.20e+02
##  [3,]  -1.00e+30   3.24e+02
##  [4,]  -1.00e+30   4.60e+02
##  [5,]   3.45e+02   4.20e+02
##  [6,]   2.52e+02   3.24e+02
##  [7,]  -1.00e+30   7.80e+02
##  [8,]  -1.00e+30   4.80e+02
##  [9,]   2.04e+02   1.00e+30
```

```
get.sensitivity.rhs(lprec)$duals
```

```
##  [1]    0    0    0   12   20   60    0    0    0  -12   84    0    0  -24  -40
## [16]    0    0 -360 -120    0
```

```
### 4.Formulate the dual of the above problem and solve it. Does the solution agree w
ith what you observed for the primal problem?
```

min Z= 750 y1 + 900 y2 + 450 y3 + 13000 y4 + 12000 y5 + 5000 y6 + 900 y7 + 1200 y8 + 750 y9 + 0 y11 + 0 y12
subject to y1 + 20y4 + y7 + 6 y10 + 3y11 + 450y12 ≥ 420 y1 + 15y4 + y7 + 6y10 + 3y11 ≥ 360 y1 + 12y4 + y7 + 6y10 + 3y11 ≥ 300 y2 + 20y5 + y8 - 5y10 ≥ 420 y2 + 15y5 + y8 - 5y10 ≥ 360 y2 + 12y5 + y8 - 5y10 ≥ 300 y3 + 20y6 + y9 - 5y11 ≥ 420 y3 + 15y6 + y9 - 5y11 ≥ 360 y3 + 12y6 + y9 - 5y11 ≥ 300

y1,y2,y3,y4,y5,y6,y7,y8,y9 ≥0 and y10,y11, y12 unrestricted in sign

```
Dual <- make.lp(0,12)

set.objfn(Dual, c(750,900,450,
                  13000,12000,5000,
                  900,1200,750,
                  0,0,0))
lp.control(Dual, 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"
```

```
add.constraint(Dual ,c(1,0,0,20,0,0,1,0,0,900,0,450), ">=", 420)
add.constraint(Dual ,c(0,1,0,0,20,0,1,0,0,-750,450,0), ">=", 420)
add.constraint(Dual ,c(0,0,1,0,0,20,1,0,0,0,-900,-750), ">=", 420)
add.constraint(Dual ,c(1,0,0,15,0,0,0,1,0,900,0,450), ">=", 360)
add.constraint(Dual ,c(0,1,0,0,15,0,0,1,0,-750,450,0), ">=", 360)
add.constraint(Dual ,c(0,0,1,0,0,15,0,1,0,0,-900,-750), ">=", 360)
add.constraint(Dual ,c(1,0,0,12,0,0,0,0,1,900,0,450), ">=", 300)
add.constraint(Dual ,c(0,1,0,0,12,0,0,0,1,-750,450,0), ">=", 300)
add.constraint(Dual ,c(0,0,1,0,0,12,0,0,1,0,-900,-750), ">=", 300)
Dual
```

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

```
options(scipen = 100)

solve(Dual)
```

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

```
get.objective(Dual)
```

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

```
get.variables(Dual)
```

```
##  [1]  0.0000000  0.0000000  0.0000000 12.0000000 20.0000000 60.0000000
##  [7]  0.0000000  0.0000000  0.0000000  0.2000000  0.4666667  0.0000000
```

```
get.constraints(Dual)
```

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

```
get.sensitivity.rhs(Dual)
```

```
## $duals
##  [1] 516.66667   0.00000   0.00000 177.77778 666.66667   0.00000   0.00000
##  [8] 166.66667 416.66667  55.55556  66.66667  33.33333   0.00000   0.00000
## [15]   0.00000 383.33333 355.55556 166.66667   0.00000   0.00000   0.00000
##
## $dualsfrom
##  [1]                             360 -1000000000000000019884624838656
##  [3] -1000000000000000019884624838656                             345
##  [5]                             345 -1000000000000000019884624838656
##  [7] -1000000000000000019884624838656                             258
##  [9]                             204 -1000000000000000019884624838656
## [11] -1000000000000000019884624838656 -1000000000000000019884624838656
## [13] -1000000000000000019884624838656 -1000000000000000019884624838656
## [15] -1000000000000000019884624838656                             -40
## [17]                             -60                             -24
## [19] -1000000000000000019884624838656 -1000000000000000019884624838656
## [21] -1000000000000000019884624838656
##
## $dualstill
##  [1]                           460.0 1000000000000000019884624838656.0
##  [3] 1000000000000000019884624838656.0                           420.0
##  [5]                           412.5 1000000000000000019884624838656.0
##  [7] 1000000000000000019884624838656.0                           324.0
##  [9] 1000000000000000019884624838656.0                           180.0
## [11]                           210.0                           480.0
## [13] 1000000000000000019884624838656.0 1000000000000000019884624838656.0
## [15] 1000000000000000019884624838656.0                            60.0
## [17]                            15.0                            32.0
## [19] 1000000000000000019884624838656.0 1000000000000000019884624838656.0
## [21]                             0.4
```

```
get.sensitivity.obj(Dual)
```

```
## $objfrom
##  [1]                              694.4444                              833.3333
##  [3]                              416.6667                            11222.2222
##  [5]                            11571.1384                             4800.0000
##  [7]                              516.6667                              844.4444
##  [9]                              583.3333                           -40000.0000
## [11]                           -15000.0000 -1000000000000000019884624838656.0000
##
## $objtill
##  [1] 1000000000000000019884624838656.0000000000000000
##  [2] 1000000000000000019884624838656.0000000000000000
##  [3] 1000000000000000019884624838656.0000000000000000
##  [4]                           13888.8888888888889596274
##  [5]                           12500.0000000000181899
##  [6]                            5181.8181818181818107405
##  [7] 1000000000000000019884624838656.0000000000000000
##  [8] 1000000000000000019884624838656.0000000000000000
##  [9] 1000000000000000019884624838656.0000000000000000
## [10]                               0.0000000007996926
## [11]                               0.0000000004798156
## [12]                               0.0000000003998463
```

```
get.dual.solution(Dual)
```

```
##  [1]   1.00000 516.66667   0.00000   0.00000 177.77778 666.66667   0.00000
##  [8]   0.00000 166.66667 416.66667  55.55556  66.66667  33.33333   0.00000
## [15]   0.00000   0.00000 383.33333 355.55556 166.66667   0.00000   0.00000
## [22]   0.00000
```