Jnoxon_3.R

Jason

2021-10-11

```
library(lpSolveAPI)
library(ggplot2)



WD<-setwd("C:/Users/Jason/Documents/MSBA/Quant")


#Problem 3

#Maximize Profit:
# 420X1 + 360X2 + 300X3 + 420X4 + 360X5+ 300X6 + 420X7 + 360X8+ 300X9
# Plant 1 <=750
# Plant 2 <=900
# Plant 3 <=450
# Space 1: 13000 >= 20X1 + 15X2 + 12X3
# Space 2: 12000 >= 20X1 + 15X2 + 12X3
# Space 3: 5000 >= 20X1 + 15X2 + 12X3
# X1 <= 900 max demand
# X2 <= 1200 max demand
# X3 <= 750 max demand
# X1/750 = X2/900 = X3/450

lpobj1 <- make.lp(0, 9)

set.objfn(lpobj1, c(420, 360, 300, 420, 360, 300, 420, 360, 300))

lp.control(lpobj1, 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"
```

```
add.constraint(lpobj1, c(20, 15, 12, 0, 0, 0, 0, 0, 0), "<=", 13000)
add.constraint(lpobj1, c(0, 0, 0, 20, 15, 12, 0, 0 , 0), "<=", 12000)
add.constraint(lpobj1, c(0, 0, 0, 0, 0, 0, 20, 15, 12), "<=", 5000)
add.constraint(lpobj1, c(1, 0, 0, 1, 0, 0, 1, 0, 0), "<=", 900) #prod 1 demand
add.constraint(lpobj1, c(0, 1, 0, 0, 1, 0, 0, 1, 0), "<=", 1200) #prod 2 demand
add.constraint(lpobj1, c(0, 0, 1, 0, 0, 1, 0, 0, 1), "<=", 750) #prod 3 demand
add.constraint(lpobj1, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 750) #plant 1 capacity
add.constraint(lpobj1, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 900) #plant 2 capacity
add.constraint(lpobj1, c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=", 450) #plant 3 capacity
add.constraint(lpobj1, c(900, 900, 900, -750, -750, -750, 0, 0, 0), "=", 0) #proportionality constraint
add.constraint(lpobj1, c(450, 450, 450, 0, 0, 0, -750, -750, -750), "=", 0) #proportionality constraint



lpobj1
```

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

```
#non zero
set.bounds(lpobj1, lower = c(0,0,0,0,0,0,0,0, 0), columns = c(1, 2, 3, 4, 5, 6, 7, 8, 9))

solve(lpobj1)
```

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

```
get.objective(lpobj1)
```

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

```
get.variables(lpobj1)
```

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

```
get.sensitivity.rhs(lpobj1)
```

```
## $duals
##  [1]   12.00   20.00   60.00    0.00    0.00    0.00    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]  1.122222e+04  1.150000e+04  4.800000e+03 -1.000000e+30 -1.000000e+30
##  [6] -1.000000e+30 -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] 1.388889e+04 1.250000e+04 5.181818e+03 1.000000e+30 1.000000e+30
##  [6] 1.000000e+30 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
```

```
get.sensitivity.obj(lpobj1)
```

```
## $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
```

```
#solving for dual

lpobj2 <- make.lp(0, 11)

#primal solution
# 420X1 + 360X2 + 300X3 + 420X4 + 360X5+ 300X6 + 420X7 + 360X8+ 300X9
# x1+x2+x3 <=750 Plant 1
# x4+x5+x6 <=900 plant 2
# x7+x8+x9 <=450 plant 3
# Space 1: 13000 >= 20X1 + 15X2 + 12X3
# Space 2: 12000 >= 20X4 + 15X5 + 12X6
# Space 3: 5000 >= 20X7 + 15X8 + 12X9
# X1 + X4 + X7 <= 900 max demand
# X2 + X5 + X8 <= 1200 max demand
# X3 + X6 + x9 <= 750 max demand
# (X1+x2+x3)*1/750 = (x4+x5+x6)*1/900 = (x7+x8+x9)*1/450

#dual
# 750y1 + 900y2 + 450y3 + 13000y4 + 12000y5 + 5000y6 +900y7 + 1200y8 + 750y9
# y1 + 20y4 + y7 + 900y10 - 450y11 >= 420
# y1 + 20y4 + y7 + 900y10 - 450y11 >= 360
# y1 + 20y4 + y7 + 900y10 - 450y11 >= 300
# y2 + 15y5 + y8 - 750y11 >= 420
# y2 + 15y5 + y8 - 750y11 >= 360
# y2 + 15y5 + y8 - 750y11 >= 300
# y3 + 12y6 + y9 - 750y11 >= 420
# y3 + 12y6 + y9 - 750y11 >= 360
# y3 + 12y6 + y9 - 750y11 >= 300

set.objfn(lpobj2, c(750, 900, 450, 13000, 12000, 5000, 900, 1200, 750, 0, 0))

lp.control(lpobj2, 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(lpobj2, c(1, 0, 0, 20, 0, 0, 1, 0, 0, 900, 450), ">=",420)
add.constraint(lpobj2, c(1, 0, 0, 20, 0, 0, 1, 0, 0, 900, 450), ">=",360)
add.constraint(lpobj2, c(1, 0, 0, 20, 0, 0, 1, 0, 0, 900, 450), ">=",300)
add.constraint(lpobj2, c(0, 1, 0, 0, 15, 0, 0, 1, 0, 0, -750), ">=",420)
add.constraint(lpobj2, c(0, 1, 0, 0, 15, 0, 0, 1, 0, 0, -750), ">=",360)
add.constraint(lpobj2, c(0, 1, 0, 0, 15, 0, 0, 1, 0, 0, -750), ">=",300)
add.constraint(lpobj2, c(0, 0, 1, 0, 0, 12, 0, 0, 1, 0, -750), ">=",420)
add.constraint(lpobj2, c(0, 0, 1, 0, 0, 12, 0, 0, 1, 0, -750), ">=",360)
add.constraint(lpobj2, c(0, 0, 1, 0, 0, 12, 0, 0, 1, 0, -750), ">=",300)



#rm(lpobj2)


solve(lpobj2)
```

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

```
get.objective(lpobj2)
```

```
## [1] 511000
```

```
get.variables(lpobj2)
```

```
##  [1]  0.0000000  0.0000000  0.0000000  0.0000000 28.0000000 35.0000000
##  [7]  0.0000000  0.0000000  0.0000000  0.4666667  0.0000000
```

```
#
# library(lpSolveAPI)
# lprec <- read.lp("weigelt.lp")
# solve(lprec)
# get.objective(lprec)
# get.variables(lprec)
# get.constraints(lprec)
```