Module 4 - Solve LP Model Using R

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#install.packages("lpSolveAPI")
```

Now, load the library

```
library(lpSolveAPI)
```

Let us set up the Weigelt Corporation problem. We have 9 decision variables, and 8 constraints. In the first formulation, we will directly create the objective function and constraints

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

# Creating objective function. The default is a minimization problem.
set.objfn(lprec, c(420,360,300,420,360,300,420,360,300))

# As the default is a minimization problem, we change the direction to set maximization
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"
```

```
# Add the constraints
#Capacity constraints
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)

#squarefoot storage
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
add.constraint(lprec, c(1, 0, 0, 1, 0 ,0, 1, 0, 0), "<=", 900)
add.constraint(lprec, c(0, 1, 0, 0, 1 ,0, 0, 1, 0), "<=", 1200)
add.constraint(lprec, c(0, 0, 1, 0, 0 ,1, 0, 0, 1), "<=", 750)

#percentage of capacity
add.constraint(lprec, c(900, 900, 900, -750, -750 ,-750, 0, 0, 0), "=", 0)
add.constraint(lprec, c(450, 450, 450, 0, 0 ,0, -750, -750, -750), "=", 0)

# Set bounds for variables explicitly.

set.bounds(lprec, lower = c(0, 0, 0, 0, 0, 0, 0, 0, 0), columns = c(1, 2,3,4,5,6,7,8,9))

# set variable names and name the constraints
RowNames <- c("Plant1Capacity", "Plant2Capacity", "Plant3Capacity","Plant1Storage", "Plant2Storage", "Pl
ColNames <- c("LargeProdPlant1", "MediumProdPlant1", "SmallProdPlant1", "LargeProdPlant2", "MediumProdPl
dimnames(lprec) <- list(RowNames, ColNames)

# Now, print out the model
lprec
```

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

```
# The model can also be saved to a file
write.lp(lprec, filename = "WeigeltCorp.lp", type = "lp")
```

We now solve the above LP problem

```
solve(lprec)
```

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

The output above doesn't indicate that the answer is 0, but that there was a successful solution. We now output the value of the objective function, and the variables

```
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
```

```
x <- read.lp("WeigeltCorp.lp") # create an lp object x
x                              # display x
```

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

Solve the lp model

```
solve(x)
```

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

```r
get.objective(x)        # get objective value
```

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

```r
get.variables(x)        # get values of decision variables
```

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

```r
get.constraints(x)      # get constraint RHS values
```

```
##  [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
```