QMM Assignment_Integer Programming

Anusha Banda

2023-11-19

Constructing a table displaying the projected manpower needed daily throughout the week. Here we are using library formattable to get better visualization

```r
library(formattable)
```

```
## Warning: package 'formattable' was built under R version 4.3.2
```

```r
data_1 <- data.frame(
  Day = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"),
  Workers_Required = c(18, 27, 22, 26, 25, 21, 19)
)

formattable(data_1, align = c("l", "c"), list(
  Workers_Required = color_tile("white", "lightblue")
))
```

Day	Workers_Required
Sunday	18
Monday	27
Tuesday	22
Wednesday	26
Thursday	25
Friday	21
Saturday	19

Package handlers at AP are ensured a five-day workweek with two successive days of rest. The fundamental wage for these handlers amounts to $750 per week. Employees engaged on either Saturday or Sunday acquire an extra $25 per day. The available shifts and compensations for package handlers are:

#Creating a table to show the possible shifts and salaries for package handlers

```r
library(formattable)

data_2 <- data.frame(
  Shift = c(1, 2, 3, 4, 5, 6, 7),
  Days_Off = c("Sunday and Monday", "Monday and Tuesday", "Tuesday and Wednesday",
               "Wednesday and Thursday", "Thursday and Friday", "Friday and Saturday",
               "Saturday and Sunday"),
  Wage = c(775, 800, 800, 800, 800, 775, 750)
)

formattable(data_2, align = c("c", "l", "c"), list(
  Wage = color_tile("white", "lightgreen")
))
```

Shift	Days_Off	Wage
1	Sunday and Monday	775
2	Monday and Tuesday	800
3	Tuesday and Wednesday	800
4	Wednesday and Thursday	800
5	Thursday and Friday	800
6	Friday and Saturday	775
7	Saturday and Sunday	750

Question: The manager wants to keep the total wage expenses as low as possible while ensuring that there are sufficient number of workers available each day. Formulate and solve the problem. What was the total cost? How many workers are available each day?

Problem Formulation: To address this scenario, let's first define the decision variables: w1 - Quantity of labor assigned to Shift 1

w2 - Quantity of labor assigned to Shift 2

w3 - Quantity of labor assigned to Shift 3

w4 - Quantity of labor assigned to Shift 4

w5 - Quantity of labor assigned to Shift 4

w6 - Quantity of labor assigned to Shift 6

w7 - Quantity of labor assigned to Shift 7

Objective Function:

Here in this scenario manager wants to minimize the labor costs, so it can be expressed as below,

Minimize =775w1+800w2+800w3+800w4+800w5+775w6+750w7

Constraints:

Establishing conditions that correspond to the seven decision variables across all weekdays.

Sunday:w2+w3+w4+w5+w6≥18

Monday:w3+w4+w5+w6+w7≥27

Tuesday:w1+w4+w5+w6+w7≥22

Wednesday:w1+w2+w5+w6+w7≥26

Thursday:w1+w2+w3+w6+w7≥25

Friday:w1+w2+w3+w4+w7≥21

Saturday:w1+w2+w3+w4+w5≥19

Non-negativity of the decision variables:

w1≥0,w2≥0,w3≥0,w4≥0,w5≥0,w6≥0,w7≥0

Solving the problem in R markdown.

Here we should load the required libraries. please see below,

```r
library(lpSolveAPI)
```

#Creating an Linear Programming problem instance with 7 constraints and 7 decision variables.Here we are initializing LP without constraints initially

```r
lp <- make.lp(0, 7)
```

Determining the coefficients that correspond to the objective function across the seven decision variables.

```r
set.objfn(lp, c(775,800,800,800,800,775,750))
```

Establishing the Objective sense to 'min'.

```r
lp.control(lp, 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"
```

Given the nature of this problem demanding an integer programming solution, it necessitates specifying the variable type as integers, ensuring solely whole numbers are utilized, not fractions. Adding the constraints to the LP problem.

```r
constraints <- list(
  c(0, 1, 1, 1, 1, 1, 0),
  c(0, 0, 1, 1, 1, 1, 1),
  c(1, 0, 0, 1, 1, 1, 1),
  c(1, 1, 0, 0, 1, 1, 1),
  c(1, 1, 1, 0, 0, 1, 1),
  c(1, 1, 1, 1, 0, 0, 1),
  c(1, 1, 1, 1, 1, 0, 0)
)

rhs <- c(18, 27, 22, 26, 25, 21, 19)

for (i in 1:7) {
  add.constraint(lp, constraints[[i]], ">=", rhs[i])
}
```

Specifying the lower bounds for decision variables, guaranteeing non-negativity constraints are maintained.

```r
set.bounds(lp, lower = rep(0, 7))
```

Naming the constraints to uniquely identify and reference each individual constraint.

```r
lp.rownames<-c("Sunday","Monday","Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
dimnames(lp) = list(lp.rownames,(1:7))
```

Specifying the decision variables to exclusively utilize integers.

```r
set.type(lp, columns = 1:7, type = "integer")
```

Solving LP problem

```r
solve(lp)
```

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

Retrieving optimal objective value

```r
get.objective(lp)
```

```
## [1] 25675
```

Retrieving values of decision variables at the optimal solution

```r
round(get.variables(lp))
```

```
## [1]  2  4  5  0  8  1 13
```

Assigning row and column names after the LP problem has constraints. By utilizing the variable from the LP model

```r
Table = matrix(c(0,4,5,0,8,1,0,0,0,5,0,8,1,13,2,0,0,0,8,1,13,2,4,0,0,8,1,13,2,4,5,0,0,1,13,2,3,4,0,0,0,13,2,4,5,
0,8,0,0),ncol=7,byrow=TRUE)

colnames(Table) = c("Shift1", "Shift2", "Shift3", "Shift4", "Shift5", "Shift6", "Shift7")

row.names(Table) = c('Sunday', 'Monday', 'Tuesday','Wednesday','Thursday','Friday','Saturday')

Table
```

```
##           Shift1 Shift2 Shift3 Shift4 Shift5 Shift6 Shift7
## Sunday         0      4      5      0      8      1      0
## Monday         0      0      5      0      8      1     13
## Tuesday        2      0      0      0      8      1     13
## Wednesday      2      4      0      0      8      1     13
## Thursday       2      4      5      0      0      1     13
## Friday         2      3      4      0      0      0     13
## Saturday       2      4      5      0      8      0      0
```

Displaying LP details

```r
lp
```

```
## Model name:
##               1    2    3    4    5    6    7
## Minimize    775  800  800  800  800  775  750
## Sunday        0    1    1    1    1    1    0  >=  18
## Monday        0    0    1    1    1    1    1  >=  27
## Tuesday       1    0    0    1    1    1    1  >=  22
## Wednesday     1    1    0    0    1    1    1  >=  26
## Thursday      1    1    1    0    0    1    1  >=  25
## Friday        1    1    1    1    0    0    1  >=  21
## Saturday      1    1    1    1    1    0    0  >=  19
## Kind        Std  Std  Std  Std  Std  Std  Std
## Type        Int  Int  Int  Int  Int  Int  Int
## Upper       Inf  Inf  Inf  Inf  Inf  Inf  Inf
## Lower         0    0    0    0    0    0    0
```

Displayed in the table below is the workforce availability for each day, arranged according to the shift schedule intended to decrease the total wage outlay.

```r
rowSums(Table)
```

```
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##        18        27        24        28        25        22        19
```