BA64018_Assignment6

Ruthvick Bulagakula

2023-11-20

Loading required libraries.

```
library(lpSolve)
library(lpSolveAPI)
```

Problem

AP is a package delivery service that guarantees overnight delivery in the continental United States. The company has hubs in major cities and airports throughout the country. Packages are received at hubs before being shipped to intermediate hubs or their final destination. The manager of the AP hub in Cleveland is concerned about labor costs and wants to know the most efficient way to schedule employees. The hub is open seven days a week, and the number of packages handled varies from day to day. The table below estimates the number of workers required on each day of the week.

```
Week_Days = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
Workers_required = c(20,25,22, 28, 25,22, 20)
Worker_per_day = data.frame(Week_Days, Workers_required)
Worker_per_day
```

```
##   Week_Days Workers_required
## 1    Sunday               20
## 2    Monday               25
## 3   Tuesday               22
## 4 Wednesday               28
## 5  Thursday               25
## 6    Friday               22
## 7  Saturday               20
```

AP guarantees package handlers a five-day work week with two consecutive days off. The weekly starting salary for handlers is $750. Workers who work on Saturday or Sunday are paid an extra $20 per day. Package handlers may work in a variety of shifts and earn a variety of wages.

```
Shifts = c(1,2,3,4,5,6,7)
Week_Days_off = c("Sunday & Monday", "Monday & Tuesday", "Tuesday & Wednesday", "Wednesday & Thursday",
Wages = c(770, 790, 790, 790, 790, 770, 750)
shift_wage = data.frame(Shifts, Week_Days_off, Wages)
shift_wage
```

```
##   Shifts        Week_Days_off Wages
## 1      1      Sunday & Monday   770
## 2      2     Monday & Tuesday   790
## 3      3  Tuesday & Wednesday   790
## 4      4 Wednesday & Thursday   790
## 5      5    Thursday & Friday   790
## 6      6    Friday & Saturday   770
## 7      7    Saturday & Sunday   750
```

Questions

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.

1. Formulate the problem.
2. Solve the problem in R markdown.
3. Find the total cost and the number of workers available each day.

Hint: The number of available workers each day can exceed, but cannot be below the required amount.

1. Problem Formulation

Let us assume that,

Number of people working in shift 1

$$x_1$$

Number of people working in shift 2

$$x_2$$

Number of people working in shift 3

$$x_3$$

Number of people working in shift 4

$$x_4$$

Number of people working in shift 5

$$x_5$$

Number of people working in shift 6

$$x_6$$

Number of people working in shift 7

$$x_7$$

People working on Sunday

$$x_2 + x_3 + x_4 + x_5 + x_6$$

People working on Monday

$$x_3 + x_4 + x_5 + x_6 + x_7$$

People working on Tuesday

$$x_4 + x_5 + x_6 + x_7 + x_1$$

People working on Wednesday

$$x_5 + x_6 + x_7 + x_1 + x_2$$

People working on Thursday

$$x_6 + x_7 + x_1 + x_2 + x_3$$

People working on Friday

$$x_7 + x_1 + x_2 + x_3 + x_4$$

People working on saturday

$$x_1 + x_2 + x_3 + x_4 + x_5$$

$$MinZ = 770x_1 + 790x_2 + 790x_3 + 790x_4 + 790x_5 + 770x_6 + 750x_7$$

Constraints:

People working on Sunday can work more than the required amount, but not less.

$$x_2 + x_3 + x_4 + x_5 + x_6 >= 20$$

People working on Monday can work more than the required amount, but not less.

$$x_3 + x_4 + x_5 + x_6 + x_7 >= 25$$

People working on Tuesday can work more than the required amount, but not less.

$$x_4 + x_5 + x_6 + x_7 + x_1 >= 22$$

People working on Wednesday can work more than the required amount, but not less.

$$x_5 + x_6 + x - 7 + x_1 + x_2 >= 28$$

People working on Thursday can work more than the required amount, but not less.

$$x_6 + x_7 + x_1 + x_2 + x_3 >= 25$$

People working on Friday can work more than the required amount, but not less.

$$x_7 + x_1 + x_2 + x_3 + x_4 >= 22$$

People working on Saturday can work more than the required amount, but not less.

$$x_1 + x_2 + x_3 + x_4 + x_5 >= 20$$

Non-negativity of constraints:

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7 >= 0$$

2. Solve the problem in R markdown.

Creating a LP problem with 7 constraints and 7 decision variables.

```
solvelp = make.lp(7, 7)
```

solvelp is an object that represents the LP problem.

This line establishes the seven decision variables' objective function coefficients. Here, a vector containing the coefficients is given. There should be a consistent ordering of the variables. The objective function and constraints should follow a consistent order.

```
set.objfn(solvelp, c(770, 790, 790, 790, 790, 770, 750))
```

By setting the objective sense to "min" this line's aim is to maximize the objective function.

```
lp.control(solvelp, 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"
```

Adding constraints to the LP problem.

```r
set.row(solvelp,1,c(0,1,1,1,1,1,0))

set.row(solvelp,2,c(0,0,1,1,1,1,1))

set.row(solvelp,3,c(1,0,0,1,1,1,1))

set.row(solvelp,4,c(1,1,0,0,1,1,1))

set.row(solvelp,5,c(1,1,1,0,0,1,1))

set.row(solvelp,6,c(1,1,1,1,0,0,1))

set.row(solvelp,7,c(1,1,1,1,1,0,0))
```

Setting the right hand side values for the constraints in LP problem.

```r
right = c(20, 25, 22, 28, 25, 22, 20)
set.rhs(solvelp,right)
```

Setting the directions(Types) and bounds to the LP problem.

```
set.constr.type(solvelp,c(">=", ">=", ">=", ">=", ">=", ">=", ">="))
set.bounds(solvelp,lower = rep(0,7))
```

```
set.type(solvelp, 1:7 ,type = c("integer"))
```

Assigning rownames to the constraints.

```
lp.rownames = c("1", "2", "3", "4", "5", "6", "7")
```

Assigning names to the decision variables. p stands for plus and m for minus.

```
lp.colnames = c("x1", "x2", "x3", "x4", "x5", "x6", "x7")
```

Solving the Lp problem using the constraints and objective function.

```
solve(solvelp)
```

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

Getting the value of objective function.

```
get.objective(solvelp)
```

```
## [1] 25550
```

Getting the values of decision variables.

```
get.variables(solvelp)
```

```
## [1]  2  6  4  0  8  2 11
```

Getting the values of constraints.

```
get.constraints(solvelp)
```

```
## [1] 20 25 23 29 25 23 20
```

3.Find the total cost and the number of workers available each day

The total cost of the optimal solution is $25,550.

The number of workers available for each day is determined based on the values of the decision variables:

Number of people working on Sunday = 20

Number of people working on Monday = 25

Number of people working on Tuesday = 23

Number of people working on Wednesday = 29

Number of people working on Thursday = 25

Number of people working on Friday = 23

Number of people working on Saturday = 20

In summary, the solution provides an optimal schedule for workers that minimizes the total cost while meeting the required number of workers for each day. The interpretation involves understanding the formulated LP problem, the process of solving it, and extracting meaningful information from the results.