Untitled

Riad Alharithi

11/29/2021

1. PM
2. Eng,
3. Surveyor,
4. Designer

i is set of people {employee 1, employee 2, }

j is set of projects

k is set of classifications {PM, Engineer, Designer, Surveyor}

Decision variables:

1. $X_{i,j,k}$ Continuous: Hours that an employee i work in functional class k in project j
2. $Y_{i,j,k} = 0, 1$ Binary variable, If employee i is assigned to classification k for project j then $y_{i,j,k}$is 1 otherwise it is 0.

```
## $A_i,_k$ = ${0, 1}$ binary, can employee $i$ do task $k$
```

3. $S_{i,k}$ =1 if employee is qualified to do task k
4. $P_{i,k}$: Profit, Hourly rate for employee i classification k in project j * 10% *
5. E_j : project budget = Not to Exceed (NTE) cost per project
6. $R_{i,k}$: billing rate for i when performing task classification k
7. $T_{j,k}$ is Total hours per project for task classification k

Objective Function:

$$\text{Maximize } \sum_i \sum_j \sum_k X_{i,j,k} P_{i,k}$$

Constraints

Utilization:total hours assigned to a staff who classified as engineer/ designer/ surveyor should not exceed 1664 hours per year.

1. $\sum_{j=1}^{j=NProj} \sum_{k=1}^{k=3} X_{i,j,k} \leq 1664$ hours $\forall i$
2. $\sum_{j=1}^{j=NProj} \sum_{k=1}^{k=3} R_{i,k} \cdot X_{i,j,k} \leq E_j \ \ \forall j$

3. $X_{i,j,k} \leq BigM * Y_{i,j,k} \forall i,j,k$
4. $Y_{i,j,k} \leq S_{i,k} \forall ijk$
5. $\sum_{i=1}^{i=NP} X_{i,j,k} \leq T_{j,k} \forall i$

Including the libraries:

```
library(ompr, quietly = TRUE)
library(magrittr, quietly = TRUE)
library(pander, quietly = TRUE)
library(ROI, quietly = TRUE)
```

```
## ROI: R Optimization Infrastructure
```

```
## Registered solver plugins: nlminb, glpk, lpsolve, neos, symphony.
```

```
## Default solver: auto.
```

```
library(ROI.plugin.glpk, quietly = TRUE)
library(ompr.roi, quietly = TRUE)
library(pander, quietly = TRUE)
library(TRA)
library(Benchmarking, quietly=TRUE)
library(ROI.plugin.glpk)
library(ROI.plugin.lpsolve)
```

```
##
## Attaching package: 'ROI.plugin.lpsolve'
```

```
## The following objects are masked from 'package:lpSolveAPI':
##
##     read.lp, write.lp
```

```
library(ROI.plugin.neos)
library(ROI.plugin.symphony)
#library(ROI.plugin.cplex)
```

```
Classification<-c("PM", "Engineer", "Designer", "Surveyor")
##Classification of staff
BillingRate <- c(200, 186, 120, 135)
Billing <-matrix(c(200, 186, 120, 135), ncol = 4, nrow = 4,
                     dimnames = c(list(c("P1", "P2", "P3", "P4")),
                                  list(c("PM", "Engineer", "Designer", "Surveyor")))
                     )
## Billing Rates per above classification in order
MaxHours<-1664
## 2080 hours per year multiplied by 80% utilization per staff = 1664
## we will need to change this to be a variable so each class has its ulitzation factor
ProfitPerStaff<-0.1*BillingRate
print (ProfitPerStaff)
```

```
## [1] 20.0 18.6 12.0 13.5
```

```
## profit is assumed 10% per the hourly rate
Staff<-matrix(c("St1","St2", "St3", "St4", "St5", "St6", "St7", "St8", "St9", "St10"), ncol = 10)
## These are all roster of technical staff working at the firm


ProfitPerClass<- matrix(c(200, 400, 400, 400, 600, 400, 350,
                  400, 75, 150, 300, 400, 750, 100,
                  300, 120, 300, 400, 650, 450, 400)*0.1,
                nrow = 7, ncol = 3)
ProfitPerClass
```

```
##      [,1] [,2] [,3]
## [1,]   20 40.0   30
## [2,]   40  7.5   12
## [3,]   40 15.0   30
## [4,]   40 30.0   40
## [5,]   60 40.0   65
## [6,]   40 75.0   45
## [7,]   35 10.0   40
```

```
n<-10
## Number of Staff
p<-3
##Number of projects
c<-7
## Number of classifications
```

```
model <- MIPModel() %>%
    add_variable(Vx[i,j,k], i=1:n, j=1:p, k=1:c, type= "continuous") %>%
  ## hours assigned to staff i on project j, in a classification k
    add_variable(Vy[i,j,k], i=1:n, j=1:p, k=1:c, type = "binary") %>%
  ## whether staff i is assigned to project j in a  classification k
    add_variable(S[i,k], i=1:n, k=1:c, type = "binary") %>%
  ##can staff i, do task k
    #add_variable(R[i,k], i=1:n, k=1:c, type = "continuous") %>%
  ##  hours needed for project i to perform task in classification k
    #add_variable (T[j,k], j=1:p, k=1:c, type = "continuous")%>%
    ## profit per staff
    #add_variable (ProfitPerStaff[i,k], i=1:n, k=1:c, type = "continuous")%>%
    ## Not to exceed amount of each project j
    #add_variable (E[j], j=1:p, type="continuous") %>%

    set_objective(sum_expr(ProfitPerClass[k, j] * Vy[i, j, k], i=1:n, j=1:3, k=1:c), "max") %>%

    #add_constraint (sum_expr (sum_expr( sum_expr(Vx[i,j,k], i=1:n), j=1:p), k=1:c) <=1664) %>%
    #add_constraint (sum_expr (sum_expr (R[i,k]*Vx[i,j,k],i=1:n), k=1:c) <= E[j] , j=1:p) %>%
    add_constraint (Vy[i,j,k]<=S[i,k], i=1:n, j=1:p, k=1:c)
    #add_constraint (sum_expr (Vx[i,j,k], i=1:n, j=1:p, k=1:c) <= Hours[k, j], k=1:7, j=1:3)

model
```

```
## Mixed integer linear optimization problem
```

```
## Variables:
##   Continuous: 210
##   Integer: 0
##   Binary: 280
## Model sense: maximize
## Constraints: 210
```

```
result <- solve_model(model, with_ROI(solver = "glpk"))
result
```

```
## Status: optimal
## Objective value: 7545
```