Module 11 – Integer Programming

shiva gadila

2023-11-20

```
day_workers <- c(Sunday = 20, Monday = 25, Tuesday = 22, Wednesday = 28,
Thursday = 25, Friday = 22, Saturday = 20)

workers_table <- data.frame(Day = names(day_workers), Required_Workers =
as.numeric(day_workers), row.names = NULL)

print(workers_table)
```

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

```
# Define the shift days off wage
shift_days_off_wage <- c("Sunday and Monday" = 770,
                         "Monday and Tuesday" = 790,
                         "Tuesday and Wednesday" = 790,
                         "Wednesday and Thursday" = 790,
                         "Thursday and Friday" = 790,
                         "Friday and Saturday" = 770,
                         "Saturday and Sunday" = 750)

# Create a data frame for the shift days off wage
shift_days_off_table <- data.frame(Shifts = 1:7, Days_Off =
names(shift_days_off_wage), Wage = as.numeric(shift_days_off_wage), row.names
= NULL)

# Print the table
print(shift_days_off_table)
```

```
##   Shifts               Days_Off Wage
## 1      1      Sunday and Monday  770
## 2      2     Monday and Tuesday  790
## 3      3  Tuesday and Wednesday  790
## 4      4 Wednesday and Thursday  790
## 5      5    Thursday and Friday  790
## 6      6    Friday and Saturday  770
## 7      7    Saturday and Sunday  750
```

Let the Number of Workers working on shift 1, 2, 3, 4, 5, 6 & 7 are X1, X2, X3, X4, X5, X6 & X7 respectively

QUESTION 1 :- Formulate the problem.

$$MINZ = 770X_1 + 790X_2 + 790X_3 + 790X_4 + 790X_5 + 770X_6 + 750X_7$$

Constraints would be:

$$Sunday: -X_2 + X_3 + X_4 + X_5 + X_6 \geq 20$$

$$Monday: -X_3 + X_4 + X_5 + X_6 + X_7 \geq 25$$

$$Tuesday: -X_4 + X_5 + X_6 + X_7 + X_1 \geq 22$$

$$Wednesday: -X_5 + X_6 + X_7 + X_1 + X_2 \geq 28$$

$$Thursday: -X_6 + X_7 + X_1 + X_2 + X_3 \geq 25$$

$$Friday: -X_7 + X_1 + X_2 + X_3 + X_4 \geq 22$$

$$Saturday: -X_1 + X_2 + X_3 + X_4 + X_5 \geq 20$$

$$Non-negative integers X_i\\ X_i≥0,i=1,2,3,4,5,6,7$$

```
#Creating the lp file content.
lp_Content<- "min:770x1+790x2+790x3+790x4+790x5+770x6+750x7;
Sunday: x2 + x3 + x4 + x5 + x6 >= 20;
Monday: x3 + x4 + x5 + x6 + x7 >= 25;
Tuesday: x1 + x4 + x5 + x6 + x7 >= 22;
Wednesday: x1 + x2 + x5 + x6 + x7 >= 28;
Thursday: x1 + x2 + x3 + x6 + x7 >= 25;
Friday: x1 + x2 + x3 + x4 + x7 >= 22;
Saturday: x1 + x2 + x3 + x4 + x5 >= 20;
int x1, x2, x3, x4, x5, x6, x7;"

writeLines(lp_Content, "AP_IntegerProgramming.lp")
```

QUESTION 2 :- Solve the problem in R markdown.

```
library(lpSolveAPI)
AP_Hub_cle <- read.lp("AP_IntegerProgramming.lp")
AP_Hub_cle

## Model name:
##             x1   x2   x3   x4   x5   x6   x7
## Minimize   770  790  790  790  790  770  750
## Sunday       0    1    1    1    1    1    0  >=  20
## Monday       0    0    1    1    1    1    1  >=  25
## Tuesday      1    0    0    1    1    1    1  >=  22
## Wednesday    1    1    0    0    1    1    1  >=  28
## Thursday     1    1    1    0    0    1    1  >=  25
```

```
## Friday       1    1    1    1    0    0    1  >=  22
## Saturday     1    1    1    1    1    0    0  >=  20
## 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
```

```
solve(AP_Hub_cle)
```

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

```
get.objective(AP_Hub_cle)
```

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

```
get.variables(AP_Hub_cle)
```

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

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

```
#Number of workers available and wage for each shift.
shift_data <- data.frame(
  Shift = 1:7,
  Workers_Available = c(2, 6, 4, 0, 8, 2, 11),
  Wage_Per_Worker = c(770, 790, 790, 790, 790, 770, 750)
)

# cost for each shift
shift_data$Cost = shift_data$Workers_Available * shift_data$Wage_Per_Worker

#total cost
total_cost <- sum(shift_data$Cost)

print(total_cost)
```

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

```
print(shift_data)
```

```
##   Shift Workers_Available Wage_Per_Worker Cost
## 1     1                 2             770 1540
## 2     2                 6             790 4740
## 3     3                 4             790 3160
## 4     4                 0             790    0
## 5     5                 8             790 6320
## 6     6                 2             770 1540
## 7     7                11             750 8250
```

```
paste("total cost expanse is $" ,get.objective(AP_Hub_cle))
```

```
## [1] "total cost expanse is $ 25550"
```

```
paste("Workers available on Sunday is"
,sum(get.variables(AP_Hub_cle)[c(2,3,4,5,6)]))
```

```
## [1] "Workers available on Sunday is 20"
```

```
paste("Workers available on Monday is"
,sum(get.variables(AP_Hub_cle)[c(3,4,5,6,7)]))
```

```
## [1] "Workers available on Monday is 25"
```

```
paste("Workers available on Tuesday is"
,sum(get.variables(AP_Hub_cle)[c(4,5,6,7,1)]))
```

```
## [1] "Workers available on Tuesday is 23"
```

```
paste("Workers available on Wednesday is"
,sum(get.variables(AP_Hub_cle)[c(5,6,7,1,2)]))
```

```
## [1] "Workers available on Wednesday is 29"
```

```
paste("Workers available on Thursday is"
,sum(get.variables(AP_Hub_cle)[c(6,7,1,2,3)]))
```

```
## [1] "Workers available on Thursday is 25"
```

```
paste("Workers available on Friday is"
,sum(get.variables(AP_Hub_cle)[c(7,1,2,3,4)]))
```

```
## [1] "Workers available on Friday is 23"
```

Creating a table for workers avibale on each day for clear view

```
#Number of workers available each day

workers_available <- data.frame(
  Day = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"),
  Workers_Available = c(20, 25, 23, 29, 25, 23, 20)
)

print(workers_available)
```

```
##         Day Workers_Available
## 1    Sunday                20
## 2    Monday                25
## 3   Tuesday                23
## 4 Wednesday                29
## 5  Thursday                25
## 6    Friday                23
## 7  Saturday                20
```

In constructing an Integer Programming (IP) model for this scenario, we established decision variables, an objective function, and constraints. The decision variables

correspond to the workforce assigned to each shift. The primary goal is to minimize the overall cost, encompassing both the regular wage and supplementary pay for weekend shifts. Constraints are introduced to guarantee that the workforce aligns with the daily requirements and follows the designated shift schedule. Following the resolution of the Linear Programming model, the optimal solution indicates that the minimum total wage is $25,550, and the corresponding daily worker availability is presented in the final table.