rsingav1\_4

2023-11-20

**Formulation of Integer Optimization Problem:**

**Decision variables:**

Let A1 be the number of workers assigned to shift 1

Let A2 be the number of workers assigned to shift 2

Let A3 be the number of workers assigned to shift 3

Let A4 be the number of workers assigned to shift 4

Let A5 be the number of workers assigned to shift 5

Let A6 be the number of workers assigned to shift 6

Let A7 be the number of workers assigned to shift 7

**Objective Function:**

Minimize the total wages:

775\*A1 + 800\*A2 + 800\*A3 + 800\*A4 + 800\*A5 + 775\*A6 + 750\*A7

**Subject to Constraints:**

As per the statement: The number of available workers each day can exceed, but cannot be below the required amount

A2+A3+A4+A5+A6 >=18 for Sunday

A3+A4+A5+A6+A7 >=27 for Monday

A1+A4+A5+A6+A7 >=22 for Tuesday

A1+A2+A5+A6+A7 >=26 for Wednesday

A1+A2+A3+A6+A7 >=25 for Thursday

A1+A2+A3+A4+A7 >=21 for Friday

A1+A2+A3+A4+A5 >=19 for Saturday

Non-Negative constraints:

A1, A2, A3, A4, A5, A6, A7 >=0

Integral Constraints:

A1, A2, A3, A4, A5, A6, A7 are all integers.

# Import the lpSolveAPI library  
library(lpSolveAPI)

# Decision variables:

Let A1 be the number of workers assigned to shift 1

Let A2 be the number of workers assigned to shift 2

Let A3 be the number of workers assigned to shift 3

Let A4 be the number of workers assigned to shift 4

Let A5 be the number of workers assigned to shift 5

Let A6 be the number of workers assigned to shift 6

Let A7 be the number of workers assigned to shift 7

# Objective Function:

Minimize the total wages:

775*A1 + 800*A2 + 800*A3 + 800*A4 + 800*A5 + 775*A6 + 750\*A7

# Create a new lp problem  
AP\_IP\_model = make.lp(0, 7)  
  
# Set the objective function coefficients(decision variables)  
set.objfn(AP\_IP\_model, c(775, 800, 800, 800, 800, 775, 750))

# Subject to Constraints:

A2+A3+A4+A5+A6 >=18 for Sunday

A3+A4+A5+A6+A7 >=27 for Monday

A1+A4+A5+A6+A7 >=22 for Tuesday

A1+A2+A5+A6+A7 >=26 for Wednesday

A1+A2+A3+A6+A7 >=25 for Thursday

A1+A2+A3+A4+A7 >=21 for Friday

A1+A2+A3+A4+A5 >=19 for Saturday

# Add constraints  
add.constraint(AP\_IP\_model, c(0, 1, 1, 1, 1, 1, 0), ">=", 18) # Sunday  
add.constraint(AP\_IP\_model, c(0, 0, 1, 1, 1, 1, 1), ">=", 27) # Monday  
add.constraint(AP\_IP\_model, c(1, 0, 0, 1, 1, 1, 1), ">=", 22) # Tuesday  
add.constraint(AP\_IP\_model, c(1, 1, 0, 0, 1, 1, 1), ">=", 26) # Wednesday  
add.constraint(AP\_IP\_model, c(1, 1, 1, 0, 0, 1, 1), ">=", 25) # Thursday  
add.constraint(AP\_IP\_model, c(1, 1, 1, 1, 0, 0, 1), ">=", 21) # Friday  
add.constraint(AP\_IP\_model, c(1, 1, 1, 1, 1, 0, 0), ">=", 19) # Saturday

# Non-Negative constraints:

A1, A2, A3, A4, A5, A6, A7 >=0

# Integral Constraints:

A1, A2, A3, A4, A5, A6, A7 are all integers.

# Constrain all the decision variables to integer  
set.type(AP\_IP\_model, 1:7, "integer")   
  
# Non negative constraints for all the decision variables is already defaultly added in this package.

# Solve the above integer linear programming problem  
solve(AP\_IP\_model)

## [1] 0

# Get the optimal solution  
optimal\_solution = get.variables(AP\_IP\_model)  
cat("Optimal solution:", optimal\_solution, "\n")

## Optimal solution: 2 4 5 0 8 1 13

# The optimal solution turned out to be 2 workers for Shift 1, 4 workers for Shift 2, 5 workers for Shift 3, 0 workers for Shift 4, 8 workers for Shift 5, 1 worker for Shift 6, and a whopping 13 workers for Shift 7.

# Get the total cost  
total\_cost = get.objective(AP\_IP\_model)  
cat("Total cost:", total\_cost, "\n")

## Total cost: 25675

# Minimum overall expense incurred by the company AP in terms of worker wages is 25675$ subject to the above constraints