MGT7180 DATA-DRIVEN DECISION-MAKING

$\mathbf{D^3M}$

Assignment 2

SUBMITTED BY

GROUP C

Chuong Pham: 40411407
Dhanush M S B: 40412492
Kushi K P: 40411893
Manogna B R: 40426970
Rohan Patil: 40395741
Thomas Agnew: 40316391

April 2024

Table of Contents

1. Problem-1: Submarine Communication Cables

1.1. Decision variables

Let $x_{i,j,t}$ be the length **(in kilometre)** of cable type $i \in \{\mathrm{A, B}\}$ produced by plan $j \in \{1,2\}$ in month $t \in \{\mathrm{Jan, Feb, Mar}\}$. We, therefore, have **12 decision variables**. The variable $x_{i,j,t}$ should be non-negative, i.e., $\boldsymbol{x_{i,j,t}} \in \mathbb{R}^+$.

1.2. Constraint on plant availability hours

Let us consider plant 1 in January, its available time is 1400 hours. Thus, we have the following constraint:

$$0.3x_{A,1,\mathrm{Jan}} + 0.24x_{B,1,\mathrm{Jan}} \leq 1{,}400, \quad (\text{Plant } 1 - \text{Jan})$$

where $0.3x_{A,1,\mathrm{Jan}}$ is the number of hours plant 1 is allocated to produce type-A cable in January. This is because, **a kilometre of cable type-A takes 0.3 hour to be manufactured** from plant-1.

Similarly, we can have other 5 constraints regarding the plan availability as follows:

$$0.32x_{A,2,\mathrm{Jan}} + 0.28x_{B,2,\mathrm{Jan}} \leq 3{,}000, (\text{Plant } 2 - \text{Jan})$$

$$0.3x_{A,1,\mathrm{Feb}} + 0.24x_{B,1,\mathrm{Feb}} \leq 600, \quad (\text{Plant } 1 - \text{Feb})$$

$$0.32x_{A,2,\mathrm{Feb}} + 0.28x_{B,2,\mathrm{Feb}} \leq 800, \quad (\text{Plant } 2 - \text{Feb})$$

$$0.3x_{A,1,\mathrm{Mar}} + 0.24x_{B,1,\mathrm{Mar}} \leq 2{,}000, \quad (\text{Plant } 1 - \text{Mar})$$

$$0.32x_{A,2,\mathrm{Mar}} + 0.28x_{B,2,\mathrm{Mar}} \leq 600, \quad (\text{Plant } 2 - \text{Mar})$$

1.3. Goals

There are **6 demands** associated with the months (Jan, Feb, and Mar) and the cable types (A and B). Since the production manager knows that they do not have enough resources to meet all demand, we can treat these demands as goals as follows:

G1: $x_{A,1,\mathrm{Jan}} + x_{A,2,\mathrm{Jan}} \geq 8{,}000 \quad (\text{Type A} - \text{Jan})$

G2: $x_{B,1,\mathrm{Jan}} + x_{B,2,\mathrm{Jan}} \geq 2{,}000 \quad (\text{Type B} - \text{Jan})$

G3: $x_{A,1,\mathrm{Feb}} + x_{A,2,\mathrm{Feb}} \geq 16{,}000 \quad (\text{Type A} - \text{Feb})$

G4: $x_{B,1,\mathrm{Feb}} + x_{B,2,\mathrm{Feb}} \geq 10{,}000 \quad (\text{Type B} - \text{Feb})$

G5: $x_{A,1,\mathrm{Mar}} + x_{A,2,\mathrm{Mar}} \geq 6{,}000 \quad (\text{Type A} - \text{Mar})$

G6: $x_{B,1,\mathrm{Mar}} + x_{B,2,\mathrm{Mar}} \geq 10{,}000 \quad (\text{Type B} - \text{Mar})$

1.4. Formulate a lexicographic goal program (GP)

To formulate a lexicographic GP, we first introduce deviation variables (**measured in km**) as

- $s_{i,t}^-\in \mathbb{R}^+$: underachieved variables, where $i \in \{\mathrm{A, B}\}$ and $t \in \{\mathrm{Jan, Feb, Mar}\}$.
- $s_{i,t}^+\in \mathbb{R}^+$: overachieved variables, where $i \in \{\mathrm{A, B}\}$ and $t \in \{\mathrm{Jan, Feb, Mar}\}$.

Then we need to order the goals from the most important to the least important. Let us **compare the profit (in ₽) per km of types A and B**. **Since the cable cost, holding cost, and packing cost are the same for both types A and B, we neglect them when computing the profit**. As a result, the profit (in ₽) per km of type A is approximating to

$$14 - 6.20 = 7.80.$$

While profit (in ₽) per km of type B is approximating to

$$18 - 7.80 = 10.20.$$

We can see that **cable type B is much more important than cable type A in terms of profit** $(10.20 > 7.80)$. Hence, maximize demand type B as much as possible will bring more benefit (i.e., profit) to the business, which business board certainly takes into account. This means that $\boldsymbol{G_2 \equiv G_4 \equiv G_6 \gg G_1 \equiv G_3 \equiv G_5}$**.**

Consequently, we pre-emptively solve 2 LPs for the following goal program:

$$\boldsymbol{min} \quad \mathbf{z} = \boldsymbol{s}^-_{\mathbf{B,Jan}} + \boldsymbol{s}^-_{\mathbf{B,Feb}} + \boldsymbol{s}^-_{\mathbf{B,Mar}} \gg \boldsymbol{s}^-_{\mathbf{A,Jan}} + \boldsymbol{s}^-_{\mathbf{A,Feb}} + \boldsymbol{s}^-_{\mathbf{A,Mar}}$$

s.t.

$$x_{A,1,\mathrm{Jan}} + x_{A,2,\mathrm{Jan}} + s^-_{\mathrm{A,Jan}} - s^+_{\mathrm{A,Jan}} = 8{,}000, \quad \text{for G1}$$

$$x_{B,1,\mathrm{Jan}} + x_{B,2,\mathrm{Jan}} + s^-_{\mathrm{B,Jan}} - s^+_{\mathrm{B,Jan}} = 2{,}000, \quad \text{for G2}$$

$$x_{A,1,\mathrm{Feb}} + x_{A,2,\mathrm{Feb}} + s^-_{\mathrm{A,Feb}} - s^+_{\mathrm{A,Feb}} + s^+_{\mathrm{A,Jan}} = 16{,}000, \quad \text{for G3}$$

$$x_{B,1,\mathrm{Feb}} + x_{B,2,\mathrm{Feb}} + s^-_{\mathrm{B,Feb}} - s^+_{\mathrm{B,Feb}} + s^+_{\mathrm{B,Jan}} = 10{,}000, \quad \text{for G4}$$

$$x_{A,1,\mathrm{Mar}} + x_{A,2,\mathrm{Mar}} + s^-_{\mathrm{A,Mar}} - s^+_{\mathrm{A,Mar}} + s^+_{\mathrm{A,Feb}} = 6{,}000, \quad \text{for G5}$$

$$x_{B,1,\mathrm{Mar}} + x_{B,2,\mathrm{Mar}} + s^-_{\mathrm{B,Mar}} - s^+_{\mathrm{B,Mar}} + s^+_{\mathrm{B,Feb}} = 10{,}000, \quad \text{for G6}$$

$$0.3x_{A,1,Jan} + 0.24x_{B,1,Jan} \leq 1{,}400, \quad (Plant\ 1 - Jan)$$

$$0.32x_{A,2,Jan} + 0.28x_{B,2,Jan} \leq 3{,}000, \quad (Plant\ 2 - Jan)$$

$$0.3x_{A,1,Feb} + 0.24x_{B,1,Feb} \leq 600, \quad (Plant\ 1 - Feb)$$

$$0.32x_{A,2,Feb} + 0.28x_{B,2,Feb} \leq 800, \quad (Plant\ 2 - Feb)$$

$$0.3x_{A,1,Mar} + 0.24x_{B,1,Mar} \leq 2{,}000, \quad (Plant\ 1 - Mar)$$

$$0.32x_{A,2,Mar} + 0.28x_{B,2,Mar} \leq 600, \quad (Plant\ 2 - Mar)$$

$$s^-_{i,t} \in \mathbb{R}^+, s^+_{i,t} \in \mathbb{R}^+,$$

$$\text{and } x_{i,j,\mathrm{t}} \in \mathbb{R}^+, \ i \in \{\mathrm{A,B}\}, j \in \{1,2\}, t \in \{\mathrm{Jan, Feb, Mar}\}.$$

We consider only underachievement $(s^-_{\mathrm{i,t}})$ in the objective function of GPs because we want to minimize the unmet demand as much as possible.

Note that in G3-G6, there are additional terms $s^+_{\mathrm{A,Jan}}, s^+_{\mathrm{B,Jan}}, s^+_{\mathrm{A,Feb}}, s^+_{\mathrm{B,Feb}}$. This comes from the fact that excess production of a given month can be used for the next month. **In general, they can be used in any months later** (for example, excess product in Jan can be used for Feb or Mar). However **due to the holding (inventory) cost, it is more beneficial to use the excess production in the next month.**

Remark: We incorporate both ($s^-_{\mathrm{i,t}}$ and $s^+_{\mathrm{i,t}}$) in the constraints because the lengths of cable being produced can be greater or less than the demand. We know that the limited resources cannot satisfy the demand, so we may easily remove the overachievement ($s^+_{\mathrm{i,t}}$) from the constraints. However, the productivity can be excessed the demand for type A and under the demand for type B or reverse. Moreover, in Feb, our available hours are substantially lower (600 hours for plant 1 and 800 hours for plant 2) compared to Jan (1400 hours for plant 1 and 3,000 hours for plant 2). Despite this, Feb sees the highest demand among the three months (16,000 km type A and 10,000 km type B). Additionally, type B is more crucial than type A, its demand peaks in Feb and Mar. Given these circumstances, the overachievement in the productivity of type B in Jan must happen. More precisely, take type B as an example, the maximum length of cable type B can be produced in Feb based on the availability of plant 1 is $\frac{600}{0.24} = 2{,}500$ km, and plant 2 is $\frac{800}{0.28} \approx 2{,}857$ km. In total, there is only $2{,}500 + 2{,}857 = 5{,}357$ km of cable type B can be possibly produced, while the demand of type B in Feb is 10,000 km. Hence, in order to minimize the unmet demand of type B, the production of this cable type must be excessed its demand in Jan.

1.5. Solve a lexicographic goal program (GP)

1.5.1. LP for goals G_2, G_4, G_6 (type-B based goals)

1.5.1.1. Mathematical models

$$\min \quad z = s^-_{\mathrm{B,Jan}} + s^-_{\mathrm{B,Feb}} + s^-_{\mathrm{B,Mar}}$$

$$
\begin{aligned}
\text{s.t.:}\ & x_{A,1,\mathrm{Jan}} + x_{A,2,\mathrm{Jan}} + s^-_{\mathrm{A,Jan}} - s^+_{\mathrm{A,Jan}} = 8{,}000, && \text{for G1}\\
& x_{B,1,\mathrm{Jan}} + x_{B,2,\mathrm{Jan}} + s^-_{\mathrm{B,Jan}} - s^+_{\mathrm{B,Jan}} = 2{,}000, && \text{for G2}\\
& x_{A,1,\mathrm{Feb}} + x_{A,2,\mathrm{Feb}} + s^-_{\mathrm{A,Feb}} - s^+_{\mathrm{A,Feb}} + s^+_{\mathrm{A,Jan}} = 16{,}000, && \text{for G3}\\
& x_{B,1,\mathrm{Feb}} + x_{B,2,\mathrm{Feb}} + s^-_{\mathrm{B,Feb}} - s^+_{\mathrm{B,Feb}} + s^+_{\mathrm{B,Jan}} = 10{,}000, && \text{for G4}\\
& x_{A,1,\mathrm{Mar}} + x_{A,2,\mathrm{Mar}} + s^-_{\mathrm{A,Mar}} - s^+_{\mathrm{A,Mar}} + s^+_{\mathrm{A,Feb}} = 6{,}000, && \text{for G5}\\
& x_{B,1,\mathrm{Mar}} + x_{B,2,\mathrm{Mar}} + s^-_{\mathrm{B,Mar}} - s^+_{\mathrm{B,Mar}} + s^+_{\mathrm{B,Feb}} = 10{,}000, && \text{for G6}\\
& 0.3x_{A,1,Jan} + 0.24x_{B,1,Jan} \le 1{,}400, (Plant\ 1 - Jan)\\
& 0.32x_{A,2,Jan} + 0.28x_{B,2,Jan} \le 3{,}000, (Plant\ 2 - Jan)\\
& 0.3x_{A,1,Feb} + 0.24x_{B,1,Feb} \le 600, (Plant\ 1 - Feb)\\
& 0.32x_{A,2,Feb} + 0.28x_{B,2,Feb} \le 800, (Plant\ 2 - Feb)\\
& 0.3x_{A,1,Mar} + 0.24x_{B,1,Mar} \le 2{,}000, (Plant\ 1 - Mar)\\
& 0.32x_{A,2,Mar} + 0.28x_{B,2,Mar} \le 600, (Plant\ 2 - Mar)\\
& s^-_{i,t} \in \mathbb{R}^+, s^+_{i,t} \in \mathbb{R}^+,\\
& \text{and } x_{i,j,\mathrm{t}} \in \mathbb{R}^+,\ i \in \{\mathrm{A,B}\}, j \in \{1,2\}, t \in \{\mathrm{Jan, Feb, Mar}\}.
\end{aligned}
$$

1.5.1.2. Implementation of LP for goal G2-G4-G6

```r

# GROUP C - Problem 1 - Submarine Communication Cables
################################################################################
# Import the ompr required packages.
library(dplyr)
library(ROI)
library(ROI.plugin.glpk)
library(ompr)
library(ompr.roi)

######################################################################
               # Type-B-based Goals (G2-G4-G6) #
######################################################################
type_B_goals <- MIPModel() %>%
  #-----------------------------------------
# The Decision Variables:
# x[i, j, t]: production of cable type i in plant j during month t (in km)
# sminus[i, t]: underachievement for cable type i in month t (in km)
# splus[i, t]: over-achievement for cable type i in month t (in km)

# Decision Variables

  add_variable(x[i, j, t], i = 1:2, j = 1:2, t = 1:3) %>%
  add_variable(sminus[i, t], i = 1:2, t = 1:3) %>%
  add_variable(splus[i, t], i = 1:2, t = 1:3) %>%

  # Objective Function: Minimize the unmet demand of type B (sum of underachievement type B)
  set_objective(sum_expr(sminus[2, t],  t = 1:3), "min") %>%

  # Constraints
  # Plant availability (in hours)
  # Plant 1
  add_constraint(0.30 * x[1, 1, 1] + 0.24 * x[2, 1, 1] <= 1400) %>%  # January
  add_constraint(0.30 * x[1, 1, 2] + 0.24 * x[2, 1, 2] <= 600) %>%   # February
  add_constraint(0.30 * x[1, 1, 3] + 0.24 * x[2, 1, 3] <= 2000) %>%  # March
  # Plant 2
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Demand fulfillment constraints for each cable type and month (in km)
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - splus[1, 1] + sminus[1, 1] == 8000) %>%             # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - splus[2, 1] + sminus[2, 1] == 2000) %>%             # January Type-B
  #adding over-production from Jan
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - splus[1, 2] + sminus[1, 2] + splus[1, 1] == 16000) %>% # February Type-A
  #adding over-production from Jan
```

```r
  add_constraint(0.30 * x[1, 1, 2] + 0.24 * x[2, 1, 2] <= 600) %>%   # February
  add_constraint(0.30 * x[1, 1, 3] + 0.24 * x[2, 1, 3] <= 2000) %>%  # March
  # Plant 2
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Demand fulfillment constraints for each cable type and month (in km)
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - splus[1, 1] + sminus[1, 1] == 8000) %>%             # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - splus[2, 1] + sminus[2, 1] == 2000) %>%             # January Type-B
  #adding over-production from Jan
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - splus[1, 2] + sminus[1, 2] + splus[1, 1] == 16000) %>% # February Type-A
  #adding over-production from Jan
  add_constraint(x[2, 1, 2] + x[2, 2, 2] - splus[2, 2] + sminus[2, 2] + splus[2, 1] == 10000) %>% # February Type-B
  #adding over-production from Feb
  add_constraint(x[1, 1, 3] + x[1, 2, 3] - splus[1, 3] + sminus[1, 3] + splus[1, 2]== 6000) %>%   # March Type-A
  #adding over-production from Feb
  add_constraint(x[2, 1, 3] + x[2, 2, 3] - splus[2, 3] + sminus[2, 3] + splus[2, 2]== 10000) %>%  # March Type-B

  # Non-negativity constraints
  set_bounds(x[i, j, t], lb = 0, i = 1:2, j = 1:2, t = 1:3) %>%   # non-negativity
  set_bounds(sminus[i, t],lb = 0, i = 1:2, t = 1:3) %>%           # non-negativity
  set_bounds(splus[i, t],lb = 0, i = 1:2, t = 1:3) %>%            # non-negativity

  # Solve the Model
  solve_model(with_ROI(solver = "glpk", verbose = TRUE))
##########################################################################
print(type_B_goals)

# Extract and display the objective value
objective_typeB_goals <- objective_value(type_B_goals)
print(paste("Objective Value for Type-B based Goals:", objective_typeB_goals))

# Extract and display the detailed solutions
solution_x1 <- get_solution(type_B_goals, x[i, j, t])
solution_sminus1 <- get_solution(type_B_goals, sminus[i, t])   #underachievement
solution_splus1 <- get_solution(type_B_goals, splus[i, t])  #over-achievement

print("Detailed Production Plan:")
print(solution_x1)
print("Detailed underachievement:")
print(solution_sminus1)
print("Detailed overachievement:")
print(solution_splus1)
#########################################################################

```

1.5.1.3. Output of LP for goal G2-G4-G6

```
PCA_A2_MGT7215_MA_Chuong_Pham_...   Group_C_Problem3_Metaheuristic.R   Group_C_Problem1_GoalProgram.R
Source on Save                                                         Run   Source
68  solution_x1 <- get_solution(type_B_goals, x[i, j, t])
69:81  (Untitled)                                                      R Script
Console   Terminal   Background Jobs
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
+   set_bounds(sminus[i, t],lb = 0, i = 1:2, t = 1:3) %>%          # non-negativity
+   set_bounds(splus[i, t],lb = 0, i = 1:2, t = 1:3) %>%           # non-negativity
+
+   # Solve the Model
+   solve_model(with_ROI(solver = "glpk", verbose = TRUE))
<SOLVER MSG>  ----
GLPK Simplex Optimizer 5.0
12 rows, 24 columns, 40 non-zeros
      0: obj =   0.000000000e+00 inf =   5.200e+04 (6)
     11: obj =   1.702380952e+04 inf =   0.000e+00 (0)
*    15: obj =   0.000000000e+00 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
<!SOLVER MSG> ----
> ##########################################################################
> print(type_B_goals)
Status: success
Objective value: 0>
> # Extract and display the objective value
> objective_typeB_goals <- objective_value(type_B_goals)
> print(paste("Objective Value for Type-B based Goals:", objective_typeB_goals))
[1] "Objective Value for Type-B based Goals: 0"
>
> # Extract and display the detailed solutions
> solution_x1 <- get_solution(type_B_goals, x[i, j, t])
> solution_sminus1 <- get_solution(type_B_goals, sminus[i, t])   #underachievement
> solution_splus1 <- get_solution(type_B_goals, splus[i, t])  #over-achievement
>
> print("Detailed Production Plan:")
[1] "Detailed Production Plan:"
> print(solution_x1)
   variable i j t     value
1         x 1 1 1 4666.6667
7         x 2 1 1    0.0000
4         x 1 2 1 3333.3333
10        x 2 2 1 6904.7619
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  229.1667
11        x 2 2 2 2595.2381
3         x 1 1 3  380.9524
9         x 2 1 3 7857.1429
6         x 1 2 3    0.0000
12        x 2 2 3 2142.8571
```

```
PCA_A2_MGT7215_MA_Chuong_Pham_...   Group_C_Problem3_Metaheuristic.R   Group_C_Problem1_GoalProgram.R
Source on Save                                                         Run   Source
68  solution_x1 <- get_solution(type_B_goals, x[i, j, t])
69:81  (Untitled)                                                      R Script
Console   Terminal   Background Jobs
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
> # Extract and display the detailed solutions
> solution_x1 <- get_solution(type_B_goals, x[i, j, t])
> solution_sminus1 <- get_solution(type_B_goals, sminus[i, t])   #underachievement
> solution_splus1 <- get_solution(type_B_goals, splus[i, t])  #over-achievement
>
> print("Detailed Production Plan:")
[1] "Detailed Production Plan:"
> print(solution_x1)
   variable i j t     value
1         x 1 1 1 4666.6667
7         x 2 1 1    0.0000
4         x 1 2 1 3333.3333
10        x 2 2 1 6904.7619
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  229.1667
11        x 2 2 2 2595.2381
3         x 1 1 3  380.9524
9         x 2 1 3 7857.1429
6         x 1 2 3    0.0000
12        x 2 2 3 2142.8571
> print("Detailed underachievement:")
[1] "Detailed underachievement:"
> print(solution_sminus1)
  variable i t     value
1   sminus 1 1     0.000
4   sminus 2 1     0.000
2   sminus 1 2 15770.833
5   sminus 2 2     0.000
3   sminus 1 3  5619.048
6   sminus 2 3     0.000
> print("Detailed overachievement:")
[1] "Detailed overachievement:"
> print(solution_splus1)
  variable i t    value
1    splus 1 1    0.000
4    splus 2 1 4904.762
2    splus 1 2    0.000
5    splus 2 2    0.000
3    splus 1 3    0.000
6    splus 2 3    0.000
> ##########################################################################
>
```

This LP results in $s^-_{\mathrm{B,Jan}} + s^-_{\mathrm{B,Feb}} + s^-_{\mathrm{B,Mar}} = \text{objective\_typeB\_goals} = 0$ which is the optimal value of the objective function.

The output indicates that $s^-_{\mathrm{B,Jan}} = s^-_{\mathrm{B,Feb}} = s^-_{\mathrm{B,Mar}} = 0$ because of non-negativity.

1.5.2. LP for goals G_1, G_3, G_5 (type-A based goals):

1.5.2.1. Mathematical models

$$\begin{aligned}
min \quad & \mathrm{z} = s^-_{\mathrm{A,Jan}} + s^-_{\mathrm{A,Feb}} + s^-_{\mathrm{A,Mar}} \\
\text{s.t.:} \quad & x_{A,1,\mathrm{Jan}} + x_{A,2,\mathrm{Jan}} + s^-_{\mathrm{A,Jan}} - s^+_{\mathrm{A,Jan}} = 8{,}000\,, && \text{for G1} \\
& x_{B,1,\mathrm{Jan}} + x_{B,2,\mathrm{Jan}} + s^-_{\mathrm{B,Jan}} - s^+_{\mathrm{B,Jan}} = 2{,}000, && \text{for G2} \\
& x_{A,1,\mathrm{Feb}} + x_{A,2,\mathrm{Feb}} + s^-_{\mathrm{A,Feb}} - s^+_{\mathrm{A,Feb}} + s^+_{\mathrm{A,Jan}} = 16{,}000, && \text{for G3} \\
& x_{B,1,\mathrm{Feb}} + x_{B,2,\mathrm{Feb}} + s^-_{\mathrm{B,Feb}} - s^+_{\mathrm{B,Feb}} + s^+_{\mathrm{B,Jan}} = 10{,}000, && \text{for G4} \\
& x_{A,1,\mathrm{Mar}} + x_{A,2,\mathrm{Mar}} + s^-_{\mathrm{A,Mar}} - s^+_{\mathrm{A,Mar}} + s^+_{\mathrm{A,Feb}} = 6{,}000, && \text{for G5} \\
& x_{B,1,\mathrm{Mar}} + x_{B,2,\mathrm{Mar}} + s^-_{\mathrm{B,Mar}} - s^+_{\mathrm{B,Mar}} + s^+_{\mathrm{B,Feb}} = 10{,}000, && \text{for G6} \\
& \boldsymbol{s^-_{\mathrm{B,Jan}} + s^-_{\mathrm{B,Feb}} + s^-_{\mathrm{B,Mar}} = 0}, && \text{result from G2-G4-G6.} \\
& 0.3x_{A,1,Jan} + 0.24x_{B,1,Jan} \leq 1{,}400, && (Plant\ 1 - Jan) \\
& 0.32x_{A,2,Jan} + 0.28x_{B,2,Jan} \leq 3{,}000, && (Plant\ 2 - Jan) \\
& 0.3x_{A,1,Feb} + 0.24x_{B,1,Feb} \leq 600, && (Plant\ 1 - Feb) \\
& 0.32x_{A,2,Feb} + 0.28x_{B,2,Feb} \leq 800, && (Plant\ 2 - Feb) \\
& 0.3x_{A,1,Mar} + 0.24x_{B,1,Mar} \leq 2{,}000, && (Plant\ 1 - Mar) \\
& 0.32x_{A,2,Mar} + 0.28x_{B,2,Mar} \leq 600, && (Plant\ 2 - Mar) \\
& s^-_{i,t} \in \mathbb{R}^+, s^+_{i,t} \in \mathbb{R}^+, \\
& \text{and } x_{i,j,\mathrm{t}} \in \mathbb{R}^+,\ i \in \{\mathrm{A,B}\}, j \in \{1,2\}, t \in \{\mathrm{Jan, Feb, Mar}\}.
\end{aligned}$$

Note that the objective value obtained from previous LP for goals G2-G4-G6 is added as a constraint in this LP.

1.5.2.2. Implementation of LP for goal G1-G3-G5

```r
print("Detailed underachievement:")
print(solution_sminus1)
print("Detailed overachievement:")
print(solution_splus1)
#########################################################################

############################################################################
                 # Type-A-based Goals (G1-G3-G5) #
############################################################################
type_A_goals <- MIPModel() %>%
  #----------------------------------------
# Decision Variables
  add_variable(x[i, j, t], i = 1:2, j = 1:2, t = 1:3) %>%
  add_variable(sminus[i, t], i = 1:2, t = 1:3) %>%
  add_variable(splus[i, t], i = 1:2, t = 1:3) %>%

  # Objective Function: Minimize the sum of shortfalls for type A (sum of underachievement type A)
  set_objective(sum_expr(sminus[1, t],  t = 1:3), "min") %>%

  # Constraints
  # Plant availability (in hours)
  # Plant 1
  add_constraint(0.30 * x[1, 1, 1] + 0.24 * x[2, 1, 1] <= 1400) %>%  # January
  add_constraint(0.30 * x[1, 1, 2] + 0.24 * x[2, 1, 2] <= 600) %>%   # February
  add_constraint(0.30 * x[1, 1, 3] + 0.24 * x[2, 1, 3] <= 2000) %>%  # March

  # Plant 2
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Add constraints from type-B based goal results (optimal value of objective function from G2-G4-G6)
  add_constraint(sum_expr(sminus[2, t], t = 1:3) == objective_typeB_goals) %>% # from G2-G4-G6 of type_B_goals

  # Demand fulfillment constraints for each cable type and month (in km)
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - splus[1, 1] + sminus[1, 1] == 8000) %>%                # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - splus[2, 1] + sminus[2, 1] == 2000) %>%                # January Type-B
  #adding excess production from Jan - Type-A
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - splus[1, 2] + sminus[1, 2] + splus[1, 1] == 16000) %>% # February Type-A
  #adding excess production from Jan - Type-B
  add_constraint(x[2, 1, 2] + x[2, 2, 2] - splus[2, 2] + sminus[2, 2] + splus[2, 1] == 10000) %>% # February Type-B
  #adding excess production from Feb - Type-A
  add_constraint(x[1, 1, 3] + x[1, 2, 3] - splus[1, 3] + sminus[1, 3] + splus[1, 2] == 6000) %>%  # March Type-A
  #adding excess production from Feb - Type-B
  add_constraint(x[2, 1, 3] + x[2, 2, 3] - splus[2, 3] + sminus[2, 3] + splus[2, 2] == 10000) %>% # March Type-B

```


```r
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Add constraints from type-B based goal results (optimal value of objective function from G2-G4-G6)
  add_constraint(sum_expr(sminus[2, t], t = 1:3) == objective_typeB_goals) %>% # from G2-G4-G6 of type_B_goals

  # Demand fulfillment constraints for each cable type and month (in km)
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - splus[1, 1] + sminus[1, 1] == 8000) %>%                # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - splus[2, 1] + sminus[2, 1] == 2000) %>%                # January Type-B
  #adding excess production from Jan - Type-A
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - splus[1, 2] + sminus[1, 2] + splus[1, 1] == 16000) %>% # February Type-A
  #adding excess production from Jan - Type-B
  add_constraint(x[2, 1, 2] + x[2, 2, 2] - splus[2, 2] + sminus[2, 2] + splus[2, 1] == 10000) %>% # February Type-B
  #adding excess production from Feb - Type-A
  add_constraint(x[1, 1, 3] + x[1, 2, 3] - splus[1, 3] + sminus[1, 3] + splus[1, 2] == 6000) %>%  # March Type-A
  #adding excess production from Feb - Type-B
  add_constraint(x[2, 1, 3] + x[2, 2, 3] - splus[2, 3] + sminus[2, 3] + splus[2, 2] == 10000) %>% # March Type-B
```


```r

  # Non-negativity constraints
  set_bounds(x[i, j, t], lb = 0, i = 1:2, j = 1:2, t = 1:3) %>%   # non-negativity
  set_bounds(sminus[i, t],lb = 0, i = 1:2, t = 1:3) %>%           # non-negativity
  set_bounds(splus[i, t],lb = 0, i = 1:2, t = 1:3) %>%            # non-negativity

  # Solve the Model
  solve_model(with_ROI(solver = "glpk", verbose = TRUE))
################################################################################
print(type_A_goals)

# Extract and display the objective value
objective_typeA_goals <- objective_value(type_A_goals)
print(paste("Objective Value for type-A based Goals (Minimize Unmet Demand):", objective_typeA_goals))

# Extract and display the detailed solution
solution_x2 <- get_solution(type_A_goals, x[i, j, t])
solution_sminus2 <- get_solution(type_A_goals, sminus[i, t])
solution_splus2 <- get_solution(type_A_goals, splus[i, t])

print("Detailed Production Plan:")
print(solution_x2)
print("Detailed underachievement:")
print(solution_sminus2)
print("Detailed overachievement:")
print(solution_splus2)
#######################################################################################
```

1.5.2.3. Output of LP for goal G1-G3-G5

```
PCA_A2_MGT7215_MA_Chuong_Pham_...   Group_C_Problem3_Metaheuristic.R   Group_C_Problem1_GoalProgram.R
Source on Save                                                         Run   Source
101    add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
84:47  (Untitled)                                                      R Script
Console   Terminal   Background Jobs
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
+   set_bounds(sminus[i, t],lb = 0, i = 1:2, t = 1:3) %>%          # non-negativity
+   set_bounds(splus[i, t],lb = 0, i = 1:2, t = 1:3) %>%           # non-negativity
+
+   # Solve the Model
+   solve_model(with_ROI(solver = "glpk", verbose = TRUE))
<SOLVER MSG>  ----
GLPK Simplex Optimizer 5.0
13 rows, 24 columns, 43 non-zeros
      0: obj =   0.000000000e+00 inf =   5.200e+04 (6)
     14: obj =   2.138988095e+04 inf =   0.000e+00 (0)
*    16: obj =   2.091666667e+04 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
<!SOLVER MSG> ----
> ################################################################################
> print(type_A_goals)
Status: success
Objective value: 20916.67>
> # Extract and display the objective value
> objective_typeA_goals <- objective_value(type_A_goals)
> print(paste("Objective Value for type-A based Goals (Minimize Unmet Demand):", objective_typeA_goals))
[1] "Objective Value for type-A based Goals (Minimize Unmet Demand): 20916.6666666667"
>
> # Extract and display the detailed solution
> solution_x2 <- get_solution(type_A_goals, x[i, j, t])
> solution_sminus2 <- get_solution(type_A_goals, sminus[i, t])
> solution_splus2 <- get_solution(type_A_goals, splus[i, t])
>
> print("Detailed Production Plan:")
[1] "Detailed Production Plan:"
> print(solution_x2)
   variable i j t     value
1         x 1 1 1    0.0000
7         x 2 1 1 5833.3333
4         x 1 2 1 8000.0000
10        x 2 2 1 1571.4286
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  666.6667
11        x 2 2 2 2095.2381
3         x 1 1 3    0.0000
9         x 2 1 3 8333.3333
6         x 1 2 3  416.6667
12        x 2 2 3 1666.6667
```

```
PCA_A2_MGT7215_MA_Chuong_Pham_...   Group_C_Problem3_Metaheuristic.R   Group_C_Problem1_GoalProgram.R
Source on Save                                                         Run   Source
101    add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
84:47  (Untitled)                                                      R Script
Console   Terminal   Background Jobs
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
> # Extract and display the detailed solution
> solution_x2 <- get_solution(type_A_goals, x[i, j, t])
> solution_sminus2 <- get_solution(type_A_goals, sminus[i, t])
> solution_splus2 <- get_solution(type_A_goals, splus[i, t])
>
> print("Detailed Production Plan:")
[1] "Detailed Production Plan:"
> print(solution_x2)
   variable i j t     value
1         x 1 1 1    0.0000
7         x 2 1 1 5833.3333
4         x 1 2 1 8000.0000
10        x 2 2 1 1571.4286
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  666.6667
11        x 2 2 2 2095.2381
3         x 1 1 3    0.0000
9         x 2 1 3 8333.3333
6         x 1 2 3  416.6667
12        x 2 2 3 1666.6667
> print("Detailed underachievement:")
[1] "Detailed underachievement:"
> print(solution_sminus2)
  variable i t     value
1   sminus 1 1     0.000
4   sminus 2 1     0.000
2   sminus 1 2 15333.333
5   sminus 2 2     0.000
3   sminus 1 3  5583.333
6   sminus 2 3     0.000
> print("Detailed overachievement:")
[1] "Detailed overachievement:"
> print(solution_splus2)
  variable i t    value
1    splus 1 1    0.000
4    splus 2 1 5404.762
2    splus 1 2    0.000
5    splus 2 2    0.000
3    splus 1 3    0.000
6    splus 2 3    0.000
> ##########################################################################################
> 
```

This LP results in the deviation variable values as presented in Table 1.

Table 1

Deviation variable values obtained from goal program for LP goal G1-G3-G5.

Deviation Variables	Values (in km)
$s^{-}_{\mathrm{A,Jan}}$	0
$s^{+}_{\mathrm{A,Jan}}$	0
$s^{-}_{\mathrm{B,Jan}}$	0
$s^{+}_{\mathrm{B,Jan}}$	5404.762
$s^{-}_{\mathrm{A,Feb}}$	15333.33
$s^{+}_{\mathrm{A,Feb}}$	0
$s^{-}_{\mathrm{B,Feb}}$	0
$s^{+}_{\mathrm{B,Feb}}$	0
$s^{-}_{\mathrm{A,Mar}}$	5583.333
$s^{+}_{\mathrm{A,Mar}}$	0
$s^{-}_{\mathrm{B,Mar}}$	0
$s^{+}_{\mathrm{B,Mar}}$	0

Which suggests that G1, G4 and G6 are satisfied. **As expected, the inventory from excess production of type B in Jan covers the shortfall in production of this type in Feb**. The minimum underachievement of type A is around 20,916 km.

1.6. Optimize total profit.

1.6.1. Mathematical model

Based on the values of deviation variables obtained from the goal program above, we are able to solve the LP to find the maximum total profit with feasible constraints by adding these values to **the 'at least' constraints**.

The total profit is equal to the revenue minus the costs.

- **Compute the revenue**:

Since the selling price is ₽ 14.00 per kilometre of Type-A and ₽ 18.00 per kilometre of Type-B, the revenue is given by

$$\begin{aligned} \text{revenue} &= 14 \times (\text{length of type A}) + 18 \times (\text{length of type B}) \\ &= 14 \times \textstyle\sum_{j\in\{1,2\}} \sum_{t\in\{\mathrm{Jan,Feb,Mar}\}} x_{A,j,t} + 18 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\mathrm{Jan,Feb,Mar}\}} x_{B,j,\boldsymbol{t}}. \end{aligned}$$

- **Compute the cost:**

$$\text{cost} = \text{cable cost} + \text{material cost} + \text{packing cost} + \text{holding cost}.$$

For the cable costs, ₽ 10.00 per hour to produce at either plant, thus the total cable cost is

$$\text{cable cost} = 10 \times (\text{total hours})$$
$$= 10 \times (0.3 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,1,t} + 0.24 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,1,t} + 0.32 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,2,t} + 0.28 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,2,t}).$$

For material cost, since material cost is ₽ 6.20 per km of Type-A and ₽ 7.80 per km of Type-B, we have

$$\text{material cost} = 6.2 \times (\text{length of type A}) + 7.8 \times (\text{length of type B})$$
$$= 6.2 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,j,t} + 7.8 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,j,t}$$

For packing cost, since each km cost ₽ 0.46, the total packing cost is

$$\text{packing cost} = 0.46 \times (\text{total cable lengths})$$
$$= 0.46 \times \sum_{i\in\{A,B\}} \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{i,j,t}.$$

For the holding cost, since each km costs ₽ 0.20, the holding cost is

$$\text{holding cost} = 0.2 \times \sum_{i\in\{A,B\}} \sum_{t\in\{\text{Jan,Feb}\}} s_{i,t}^{+}.$$

Note that only excess products are used to calculate the inventory cost. Therefore, overproduction ($s_{i,t}^{+}$) is considered in computing this cost. The cables are delivered to customers at the end of each month. Based on this, we can determine the amount of cables that were manufactured in excess during that month. These excess cables are then stored in inventory, incurring holding costs. As there are no holding costs during the production month, **only the excess products from Jan and Feb are used to calculate the inventory cost**. The excess production in Mar will take holding cost in Apr, which is not considered in this problem.

Therefore, the total profit is

$$\text{total profit} = \text{revenue} - \text{cost}$$
$$= 14 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,j,t} + 18 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,j,t} - 10 \times \Big(0.3 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,1,t} + 0.24 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,1,t} + 0.32 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,2,t} + 0.28 \times \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,2,t}\Big) - 6.2 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{A,j,t} - 7.8 \times \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{B,j,t} - 0.46 \times \sum_{i\in\{A,B\}} \sum_{j\in\{1,2\}} \sum_{t\in\{\text{Jan,Feb,Mar}\}} x_{i,j,t} - 0.2 \times \sum_{i\in\{A,B\}} \sum_{t\in\{\text{Jan,Feb}\}} \boldsymbol{s_{i,t}^{+}},$$

where $\boldsymbol{s_{i,t}^{+}}$ for $t \in \{\text{Jan, Feb}\}$ **is the notation for the value obtained from Table 1, not a decision variable in this function.**

Remark: the profit cannot be written separated by each month. This is because the plants may produce more than the demand. In this case, if we divide the above equation to three parts for each month, the revenue is not actual revenue of relevant month.

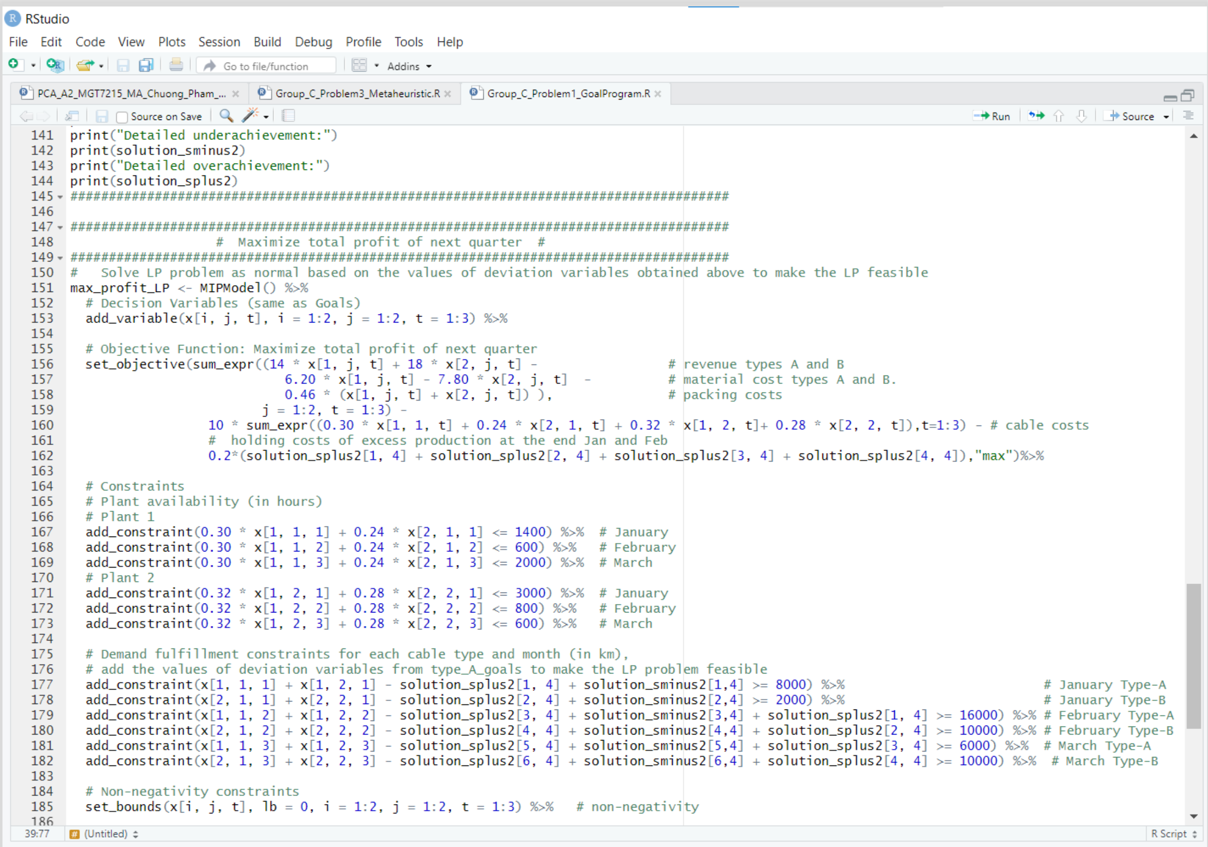
Under the production schedule we have devised in previous parts for deviation variables (Table 1), the maximum profit optimization can be formulated as follows:

$$\mathit{min} \quad \mathrm{z} = \text{total profit}$$

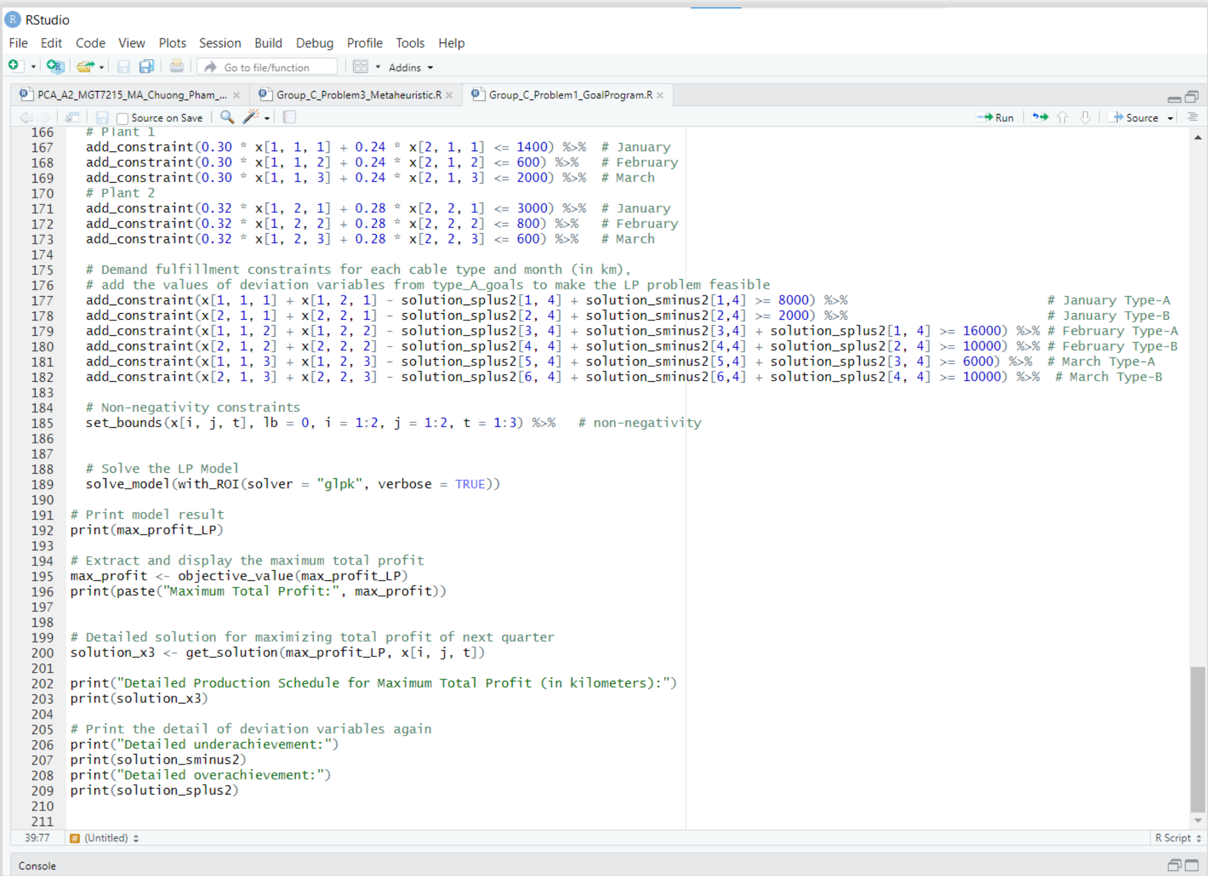
$$
\begin{aligned}
\text{s.t.} \quad & x_{A,1,\mathrm{Jan}} + x_{A,2,\mathrm{Jan}} + s^{-}_{\mathrm{A,Jan}} - s^{+}_{\mathrm{A,Jan}} \geq 8{,}000\,, && \text{for G1} \\
& x_{B,1,\mathrm{Jan}} + x_{B,2,\mathrm{Jan}} + s^{-}_{\mathrm{B,Jan}} - s^{+}_{\mathrm{B,Jan}} \geq 2{,}000, && \text{for G2} \\
& x_{A,1,\mathrm{Feb}} + x_{A,2,\mathrm{Feb}} + s^{-}_{\mathrm{A,Feb}} - s^{+}_{\mathrm{A,Feb}} + s^{+}_{\mathrm{A,Jan}} \geq 16{,}000, && \text{for G3} \\
& x_{B,1,\mathrm{Feb}} + x_{B,2,\mathrm{Feb}} + s^{-}_{\mathrm{B,Feb}} - s^{+}_{\mathrm{B,Feb}} + s^{+}_{\mathrm{B,Jan}} \geq 10{,}000, && \text{for G4} \\
& x_{A,1,\mathrm{Mar}} + x_{A,2,\mathrm{Mar}} + s^{-}_{\mathrm{A,Mar}} - s^{+}_{\mathrm{A,Mar}} + s^{+}_{\mathrm{A,Feb}} \geq 6{,}000, && \text{for G5} \\
& x_{B,1,\mathrm{Mar}} + x_{B,2,\mathrm{Mar}} + s^{-}_{\mathrm{B,Mar}} - s^{+}_{\mathrm{B,Mar}} + s^{+}_{\mathrm{B,Feb}} \geq 10{,}000, && \text{for G6} \\
& 0.3x_{A,1,Jan} + 0.24x_{B,1,Jan} \leq 1{,}400, && (Plant\ 1 - Jan) \\
& 0.32x_{A,2,Jan} + 0.28x_{B,2,Jan} \leq 3{,}000, && (Plant\ 2 - Jan) \\
& 0.3x_{A,1,Feb} + 0.24x_{B,1,Feb} \leq 600, && (Plant\ 1 - Feb) \\
& 0.32x_{A,2,Feb} + 0.28x_{B,2,Feb} \leq 800, && (Plant\ 2 - Feb) \\
& 0.3x_{A,1,Mar} + 0.24x_{B,1,Mar} \leq 2{,}000, && (Plant\ 1 - Mar) \\
& 0.32x_{A,2,Mar} + 0.28x_{B,2,Mar} \leq 600, && (Plant\ 2 - Mar) \\
& \text{and } x_{i,j,\mathrm{t}} \in \mathbb{R}^{+}, && i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{1,2\}, t \in \{\mathrm{Jan}, \mathrm{Feb}, \mathrm{Mar}\}.
\end{aligned}
$$

Note that the above optimization **is a linear program**. In addition, $\boldsymbol{s^{-}_{i,t}}$ **and** $\boldsymbol{s^{+}_{i,t}}$ **for all** $\boldsymbol{i} \in \{\mathbf{A}, \mathbf{B}\}$ **and** $\boldsymbol{t} \in \{\mathbf{Jan}, \mathbf{Feb}, \mathbf{Mar}\}$ **are constants obtained from previous lexicographic GP (Table 1).**

1.6.2. Implementation of LP to optimize total profit for the quarter.



```r
print("Detailed underachievement:")
print(solution_sminus2)
print("Detailed overachievement:")
print(solution_splus2)
######################################################################################

######################################################################################
                   #  Maximize total profit of next quarter  #
######################################################################################
#   Solve LP problem as normal based on the values of deviation variables obtained above to make the LP feasible
max_profit_LP <- MIPModel() %>%
  # Decision Variables (same as Goals)
  add_variable(x[i, j, t], i = 1:2, j = 1:2, t = 1:3) %>%

  # Objective Function: Maximize total profit of next quarter
  set_objective(sum_expr((14 * x[1, j, t] + 18 * x[2, j, t] -                 # revenue types A and B
                          6.20 * x[1, j, t] - 7.80 * x[2, j, t]  -            # material cost types A and B.
                          0.46 * (x[1, j, t] + x[2, j, t]) ),                  # packing costs
                          j = 1:2, t = 1:3) -
                10 * sum_expr((0.30 * x[1, 1, t] + 0.24 * x[2, 1, t] + 0.32 * x[1, 2, t]+ 0.28 * x[2, 2, t]),t=1:3) - # cable costs
                #  holding costs of excess production at the end Jan and Feb
                0.2*(solution_splus2[1, 4] + solution_splus2[2, 4] + solution_splus2[3, 4] + solution_splus2[4, 4]),"max")%>%

  # Constraints
  # Plant availability (in hours)
  # Plant 1
  add_constraint(0.30 * x[1, 1, 1] + 0.24 * x[2, 1, 1] <= 1400) %>%  # January
  add_constraint(0.30 * x[1, 1, 2] + 0.24 * x[2, 1, 2] <= 600) %>%   # February
  add_constraint(0.30 * x[1, 1, 3] + 0.24 * x[2, 1, 3] <= 2000) %>%  # March
  # Plant 2
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Demand fulfillment constraints for each cable type and month (in km),
  # add the values of deviation variables from type_A_goals to make the LP problem feasible
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - solution_splus2[1, 4] + solution_sminus2[1,4] >= 8000) %>%                          # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - solution_splus2[2, 4] + solution_sminus2[2,4] >= 2000) %>%                          # January Type-B
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - solution_splus2[3, 4] + solution_sminus2[3,4] + solution_splus2[1, 4] >= 16000) %>% # February Type-A
  add_constraint(x[2, 1, 2] + x[2, 2, 2] - solution_splus2[4, 4] + solution_sminus2[4,4] + solution_splus2[2, 4] >= 10000) %>% # February Type-B
  add_constraint(x[1, 1, 3] + x[1, 2, 3] - solution_splus2[5, 4] + solution_sminus2[5,4] + solution_splus2[3, 4] >= 6000) %>%  # March Type-A
  add_constraint(x[2, 1, 3] + x[2, 2, 3] - solution_splus2[6, 4] + solution_sminus2[6,4] + solution_splus2[4, 4] >= 10000) %>%  # March Type-B

  # Non-negativity constraints
  set_bounds(x[i, j, t], lb = 0, i = 1:2, j = 1:2, t = 1:3) %>%   # non-negativity
```

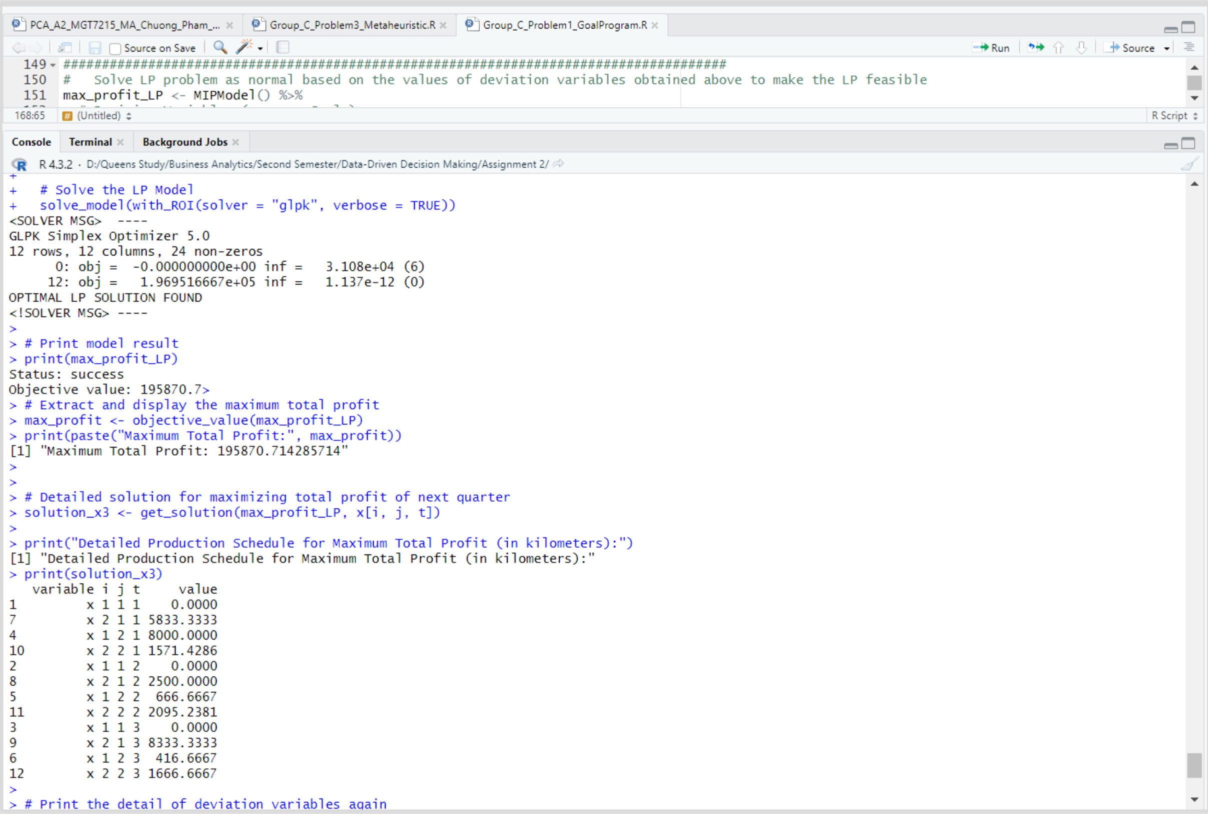



```r
  # Plant 1
  add_constraint(0.30 * x[1, 1, 1] + 0.24 * x[2, 1, 1] <= 1400) %>%  # January
  add_constraint(0.30 * x[1, 1, 2] + 0.24 * x[2, 1, 2] <= 600) %>%   # February
  add_constraint(0.30 * x[1, 1, 3] + 0.24 * x[2, 1, 3] <= 2000) %>%  # March
  # Plant 2
  add_constraint(0.32 * x[1, 2, 1] + 0.28 * x[2, 2, 1] <= 3000) %>%  # January
  add_constraint(0.32 * x[1, 2, 2] + 0.28 * x[2, 2, 2] <= 800) %>%   # February
  add_constraint(0.32 * x[1, 2, 3] + 0.28 * x[2, 2, 3] <= 600) %>%   # March

  # Demand fulfillment constraints for each cable type and month (in km),
  # add the values of deviation variables from type_A_goals to make the LP problem feasible
  add_constraint(x[1, 1, 1] + x[1, 2, 1] - solution_splus2[1, 4] + solution_sminus2[1,4] >= 8000) %>%                          # January Type-A
  add_constraint(x[2, 1, 1] + x[2, 2, 1] - solution_splus2[2, 4] + solution_sminus2[2,4] >= 2000) %>%                          # January Type-B
  add_constraint(x[1, 1, 2] + x[1, 2, 2] - solution_splus2[3, 4] + solution_sminus2[3,4] + solution_splus2[1, 4] >= 16000) %>% # February Type-A
  add_constraint(x[2, 1, 2] + x[2, 2, 2] - solution_splus2[4, 4] + solution_sminus2[4,4] + solution_splus2[2, 4] >= 10000) %>% # February Type-B
  add_constraint(x[1, 1, 3] + x[1, 2, 3] - solution_splus2[5, 4] + solution_sminus2[5,4] + solution_splus2[3, 4] >= 6000) %>%  # March Type-A
  add_constraint(x[2, 1, 3] + x[2, 2, 3] - solution_splus2[6, 4] + solution_sminus2[6,4] + solution_splus2[4, 4] >= 10000) %>%  # March Type-B

  # Non-negativity constraints
  set_bounds(x[i, j, t], lb = 0, i = 1:2, j = 1:2, t = 1:3) %>%   # non-negativity
```



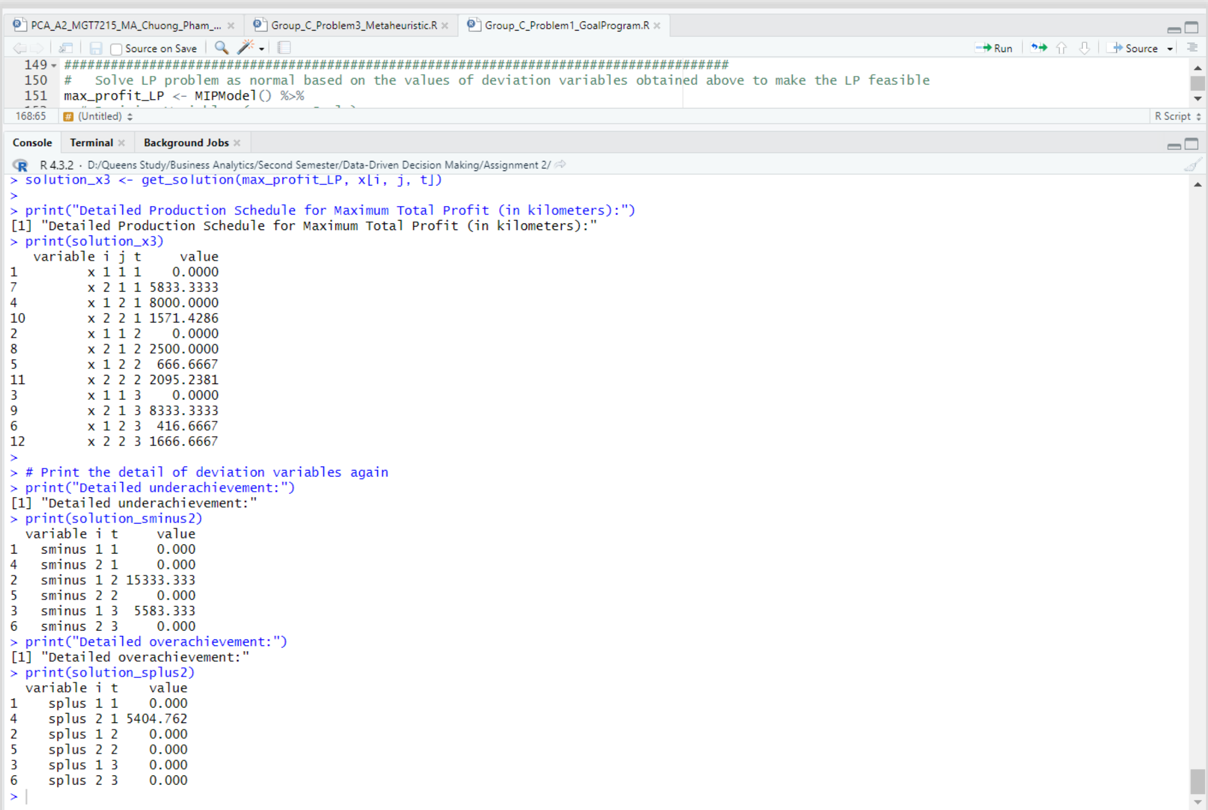
```r


  # Solve the LP Model
  solve_model(with_ROI(solver = "glpk", verbose = TRUE))

# Print model result
print(max_profit_LP)

# Extract and display the maximum total profit
max_profit <- objective_value(max_profit_LP)
print(paste("Maximum Total Profit:", max_profit))


# Detailed solution for maximizing total profit of next quarter
solution_x3 <- get_solution(max_profit_LP, x[i, j, t])

print("Detailed Production Schedule for Maximum Total Profit (in kilometers):")
print(solution_x3)

# Print the detail of deviation variables again
print("Detailed underachievement:")
print(solution_sminus2)
print("Detailed overachievement:")
print(solution_splus2)
```

1.6.3. Output of LP using ompr package in R.



```
149 ########################################################################################
150 #   Solve LP problem as normal based on the values of deviation variables obtained above to make the LP feasible
151 max_profit_LP <- MIPModel() %>%
```

```
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
+
+   # Solve the LP Model
+   solve_model(with_ROI(solver = "glpk", verbose = TRUE))
<SOLVER MSG>  ----
GLPK Simplex Optimizer 5.0
12 rows, 12 columns, 24 non-zeros
      0: obj =  -0.000000000e+00 inf =   3.108e+04 (6)
     12: obj =   1.969516667e+05 inf =   1.137e-12 (0)
OPTIMAL LP SOLUTION FOUND
<!SOLVER MSG> ----
>
> # Print model result
> print(max_profit_LP)
Status: success
Objective value: 195870.7>
> # Extract and display the maximum total profit
> max_profit <- objective_value(max_profit_LP)
> print(paste("Maximum Total Profit:", max_profit))
[1] "Maximum Total Profit: 195870.714285714"
>
>
> # Detailed solution for maximizing total profit of next quarter
> solution_x3 <- get_solution(max_profit_LP, x[i, j, t])
>
> print("Detailed Production Schedule for Maximum Total Profit (in kilometers):")
[1] "Detailed Production Schedule for Maximum Total Profit (in kilometers):"
> print(solution_x3)
   variable i j t     value
1         x 1 1 1    0.0000
7         x 2 1 1 5833.3333
4         x 1 2 1 8000.0000
10        x 2 2 1 1571.4286
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  666.6667
11        x 2 2 2 2095.2381
3         x 1 1 3    0.0000
9         x 2 1 3 8333.3333
6         x 1 2 3  416.6667
12        x 2 2 3 1666.6667
>
> # Print the detail of deviation variables again
```



```
149 ########################################################################################
150 #   Solve LP problem as normal based on the values of deviation variables obtained above to make the LP feasible
151 max_profit_LP <- MIPModel() %>%
```

```
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
> solution_x3 <- get_solution(max_profit_LP, x[i, j, t])
>
> print("Detailed Production Schedule for Maximum Total Profit (in kilometers):")
[1] "Detailed Production Schedule for Maximum Total Profit (in kilometers):"
> print(solution_x3)
   variable i j t     value
1         x 1 1 1    0.0000
7         x 2 1 1 5833.3333
4         x 1 2 1 8000.0000
10        x 2 2 1 1571.4286
2         x 1 1 2    0.0000
8         x 2 1 2 2500.0000
5         x 1 2 2  666.6667
11        x 2 2 2 2095.2381
3         x 1 1 3    0.0000
9         x 2 1 3 8333.3333
6         x 1 2 3  416.6667
12        x 2 2 3 1666.6667
>
> # Print the detail of deviation variables again
> print("Detailed underachievement:")
[1] "Detailed underachievement:"
> print(solution_sminus2)
  variable i t     value
1   sminus 1 1     0.000
4   sminus 2 1     0.000
2   sminus 1 2 15333.333
5   sminus 2 2     0.000
3   sminus 1 3  5583.333
6   sminus 2 3     0.000
> print("Detailed overachievement:")
[1] "Detailed overachievement:"
> print(solution_splus2)
  variable i t    value
1    splus 1 1    0.000
4    splus 2 1 5404.762
2    splus 1 2    0.000
5    splus 2 2    0.000
3    splus 1 3    0.000
6    splus 2 3    0.000
>
```

Note that the last ‘obj’ value shown after <SOLVER MSG> ---- is 196,951.67, while the objective value is generated as 195,870.7 due to the iterations and the significant small infeasible (inf) value 1.137×10^{-12}. Hence, the value of maximum profit we consider is the generated objective value from max_profit_LP model.

1.7. Decision

DG-Lynx should manufacture approximately 8,000 km cable type A from plant 2, 5,833 km cable type B from plant-1, and 1,571 km cable type B from plant-2 **in Jan**; around 667 km cable type A from plant-2, 2,500 km cable type B from plant-1, and 2,095 km cable type B from plant-2 **in Feb**; around 417 km cable type A from plant-2, 8,333 km cable type B from plant-1, and 1,667 km cable type B from plant-2 in **Mar** in order to achieve the maximum total profit of the next quarter of 195,870.7 in ₺.

This production schedule satisfies the demand of type-A but overachieves the demand of type-B around 5,404 km **in Jan**. In **Feb**, the demand of type-A is underachieved by around 15,333 km, while the demand of type-B is satisfied. Similarly, in **Mar**, the demand of type-A is underachieved by around 5,583 km, while the demand of type-B is satisfied.

Note that we round up the solution in km in this statement although length of cable can be fraction because this is business decision, which can be generally explained to business board for decision-making rather than the mathematical problem.

2. Problem-2: Multi-Objective Line-Balancing

2.1. Brief introduction with Methodology

To enhance gameplay in the MOLB (Multi-Objective Linear Bounded) game, several strategies borrowed from literature can be applied. Here are some effective strategies based on common gaming principles and optimization techniques:

1. **Largest Candidate Rule (LCR) with MWR / LWR**: According to Groover (2016), this technique selects the task with the longest processing time to enhance the economic efficiency of the operation. If processing times are the same, the method is improved by adding the Maximum Work Remaining (MWR) strategy, which prioritizes the task with the greatest amount of work left. In cases where processing times are equal, the LCR is modified by employing the Lowest Work Remaining (LWR) strategy, choosing the task with the least remaining work.

2. **Shortest Processing Time (SPT) with MWR / LWR**: This strategy assigns tasks based on the shortest processing times first.

3. **LCR/SPT - Kill Bridge**: This strategy merges the Least Candidate Rule with the Shortest Processing Time approach and incorporates the Kill Bridge rule, which selects tasks according to the maximum number of earlier dependencies.

4. The approach that focuses on the social objective allocates tasks in a way that evenly distributes the workload across all workstations, promoting fairness and operational efficiency.

5. For the environmental objective, tasks are assigned based on the similarity of the required tool types, as outlined by Ahmadi et al. (2023), to reduce environmental impacts.

2.2. Pareto solution for each setting

Settings/ PARETO	Solution - 1	Solution - 2	Solution - 3	Solution - 4
Setting - 1	Utilising the Largest Candidate Rule (LCR) for Pareto optimization, social objectives emerged as the primary focus with a normalised value of 0.90, followed closely by economic considerations at 0.81. Environmental aspects were less prioritised, with a value of 0.53, resulting in an overall weighted sum of 0.72, reflecting a strategy favouring social and economic factors.	The application of the Largest Candidate Rule (LCR) with a consistent tool approach yielded a prioritisation of economic objectives with a normalised value of 0.81 and environmental concerns at 0.71, while social objectives received a lower value of 0.62, as reflected in an overall weighted sum of 0.71.		

Setting - 2	By employing the Shortest Processing Time method, the Pareto analysis prioritises economic objectives with the highest normalised value of 0.80, while the environmental objective is deemed least significant at 0.54, with the weighted sum at 0.67 indicating the overall balance across all objectives.	Utilizing the LCR method with a focus on Maximum Work Remaining (MWR), the Pareto solution prioritizes the social objective with a high value of 0.91, indicating a significant focus on social factors. The environmental aspect is given the least priority with a value of 0.54, and the overall balance, as represented by the weighted sum, is at 0.67.	The Pareto solution reached using the LCR - Kilbridge method suggests a prioritisation of economic objectives highlighted as the top priority with a value of 0.80. Social and environmental objectives are also emphasised with values of 0.61 and 0.63, respectively. The aggregate weighted sum stands at 0.68, denoting the combined importance of all the objectives.	

Setting - 3	is obtained by using the method that enhances the environmental objective. More precisely, assignable tasks are assigned based on their tool-type similarity (Ahmadi et al., 2023).	is obtained by combining rules that prioritize both economic and social aspects. More precisely, since the total processing time of all tasks is 317 and the takt time is 52, the smallest number of WSs is $\left\lceil\frac{317}{52}\right\rceil = 7$. Thus, economic objective is maximised when the number of WSs is 7, and the corresponding average processing time for each WS should be 317/7 = 45.28. Based on the above observation, we use LCR heuristic rule, but for each WS, the assignment will be stopped as soon as the processing time of that WS is around 45. This guarantees high social factor and maximum economic factor.	combines economic, social, and environment aspects. More precisely, we group tasks with common tools, and stop when the processing time of the WS is around 45.	combines economic, social, and environment aspects, but focuses slightly more on environmental objective, i.e., it prioritises grouping tasks with common tools.

Setting - 4	The Pareto optimization using the Largest Candidate Rule (LCR) and Kilbridge method highlighted economic objectives with the highest priority, achieving a normalized value of 0.86. Social objectives were also emphasized with a value of 0.78, while environmental concerns received considerably less focus, marked by a normalized value of 0.39. The overall prioritization, reflected by the weighted sum of 0.67, indicates a clear preference for economic and social factors over environmental considerations.	In the optimization process utilising Shortest Processing Time, economic objectives were given the greatest importance, with a normalised value of 0.86, while social and environmental objectives received values of 0.74 and 0.44, respectively. The composite weighted sum of 0.67 indicates a systematic preference for economic over environmental concerns within the evaluated framework.		

Setting - 5	In the Pareto optimization using the Largest Candidate Rule (LCR) and Kilbridge method, economic objectives take precedence with a normalised value of 0.78. Social objectives are deemed less critical with a value of 0.40, slightly below environmental objectives at 0.44. The overall strategy achieves a moderate weighted sum of 0.58, suggesting a stronger focus on economic and environmental factors over social considerations.	Employing the Shortest Processing Time (SPT) combined with the Least Work Remaining (LWR) rule when choosing the same tool, the strategy highly values economic (0.89) and social (0.87) objectives, indicating a strong emphasis on these aspects in decision-making. Environmental objectives receive significantly less priority with a value of 0.28. The overall decision-making process, as indicated by the weighted sum of 0.68, skews towards maximising efficiency in economic and social terms while de-emphasising environmental concerns.	In the scheduling prioritisation using Shortest Processing Time, the economic objective scores the highest at 0.89. Social aspects are also valued highly at 0.77, while environmental concerns are less prioritised, reflected by a lower score of 0.39. The aggregate weighted sum is 0.69, illustrating a focus on economic and social benefits over environmental considerations.	
Setting - 6	Using the Largest Candidate Rule (LCR) method, Solution 1 prioritises economic and social objectives with normalised values of 0.92 and 0.75 respectively, while placing a lesser emphasis on environmental aspects, evidenced by a score of 0.33, leading to an overall weighted sum of 0.73.	Utilizing the Shortest Processing Time (SPT) with a similar tool type rule, the analysis prioritizes economic efficiency (0.92) over social (0.74) and environmental (0.56) aspects, with an overall weighted sum of 0.77 indicating a balanced yet economically skewed multi-criteria decision-making approach.	Solution 3 has been established as the optimal solution, with the application of Shortest Processing Time and Kill Bridge methods, achieving strong normalised values of 0.92, 0.73, and 0.61 for economic, social, and environmental objectives, respectively, and an overall weighted sum of 0.78.	

2.3. Optimal Solutions for each setting.

Setting 1: Solution 1 has been identified as the optimal solution. By applying the Largest Candidate Rule (LCR) for Pareto optimization, it prioritized social objectives with a normalized value of 0.90, with economic considerations also strong at 0.81. Environmental factors were deemed less critical, evidenced by a score of 0.53, culminating in an overall weighted sum of 0.72. This approach clearly emphasizes social and economic factors over environmental concerns.

Setting 2: Solution 3 has been identified as the optimal solution. Employing the LCR - Kilbridge method for Pareto optimization, it notably emphasizes economic objectives with a top score of 0.80. Social and environmental objectives are also given significant consideration, with scores of 0.61 and 0.63, respectively. The overall weighted sum of 0.68 indicates the balanced importance attributed to all objectives.

Setting 3: Among the four pareto solutions, Pareto solution 4 has higher weighted sum value, and hence, it is the best-found solution. The LCR is easy to use, however, to get a good result, we need to combine all economic, social, and environment aspects as the method used to obtain Pareto solution 4.

Setting 4: Solution 2 has been identified as the optimal solution. In the optimization process using Shortest Processing Time, economic objectives were prioritized, achieving a normalized value of 0.86. Meanwhile, social and environmental objectives were assigned values of 0.74 and 0.44, respectively. The total weighted sum of 0.67 reflects a strategic preference for economic objectives over environmental concerns within the assessed framework.

Setting 5: Solution 3 has been identified as the optimal solution, with the scheduling prioritization using Shortest Processing Time yielding the highest economic score at 0.89. Social aspects are also highly valued, achieving a score of 0.77, while environmental concerns are less prioritized, reflected by a lower score of 0.39. The overall weighted sum stands at 0.69, demonstrating a focus on economic and social benefits over environmental considerations.

Setting 6: Solution 3 has been discerned as the optimal solution, presenting a strong emphasis on economic goals with a score of 0.92, alongside significant consideration for social and environmental objectives, scoring 0.73 and 0.61 respectively, culminating in an aggregate weighted sum of 0.78.

As experienced, the LCR is easy to use, however, to get a good result, we need to combine all economic, social, and environment aspects as the method used to obtain Pareto solution 4 of setting-3.

2.4. Analytical Hierarchy Process (AHP) Results

In the context of Multi-Criteria Decision-Making (MCDM), our 3x3 comparison matrix quantitatively evaluates the relative importance of three decision-making criteria.

1	**3**	**5**
1/3	**1**	**4**
1/5	**1/4**	**1**

Fig. 1. 3x3 Comparison matrix.

Matrix Interpretation:

Economic Criteria: Baseline importance (value 1), weakly more important than Social (value 3), strongly more important than Environmental (value 5).
Social Criteria: Weakly less important than Economic (value 1/3), equally important to itself (value 1), moderately more important than Environmental (value 4).
Environmental Criteria: Strongly less important than Economic (value 1/5), moderately less important than Social (value 1/4), equally important to itself (value 1).

Matrix Benefits:

Structured Decision-Making: Quantifies relative importance of criteria for systematic decision-making.
Consistency: Maintains consistent evaluations across complex criteria.
Flexibility: Adaptable to various scenarios or priorities.
Quantitative Analysis: Converts qualitative assessments into objective criteria weights.
Improved Communication: Facilitates clearer stakeholder communication and consensus.

Based on the analysis of the Multi Objective Line Balancing Problem using criteria and alternatives prioritization with a 3×3 and 6×6 comparison matrix, here is a concise interpretation of the results suitable for inclusion in an academic report

a) Criteria Priorities:

- Environmental factors were given the highest priority with a weight of 0.619, indicating a significant emphasis on sustainability in the line balancing.
- Social aspects followed with a weight of 0.284, suggesting a strong but secondary focus on social impacts.
- Economic considerations were weighted the least at 0.096, highlighting less emphasis on economic factors compared to environmental and social criteria.

b) Consistency of Assignments:

- The consistency ratio of the criteria priorities was 0.075, which is well below 0.1, indicating a highly consistent judgment in the prioritization of criteria.

c) Alternatives Priorities:

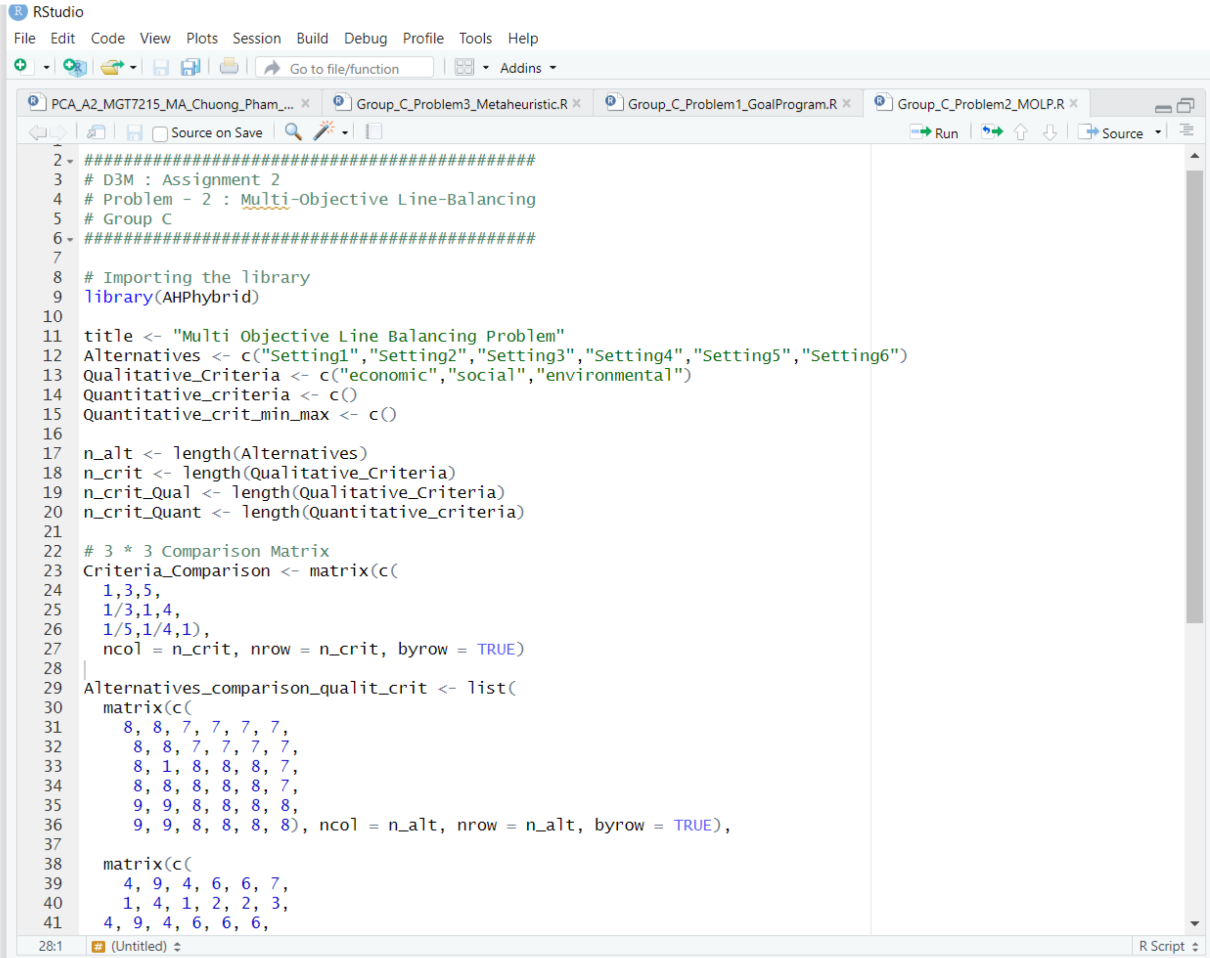
- Under the environmental criterion, the assignments were inconsistent with a consistency ratio of 6.422. This suggests that the judgments for setting priorities under this criterion might require re-evaluation or adjustment.
- Similar inconsistency issues were noted in the social (consistency ratio: 3.006) and economic (consistency ratio: 2.628) criteria, indicating potential discrepancies in judgments or the need for clearer decision-making criteria.

d) Final Results:

The final global index ranked Setting1, Setting3, and Setting6 as the top three settings, with scores of 0.175, 0.173, and 0.172, respectively. This reflects a balanced consideration of all criteria, albeit with a slightly higher weight towards environmental factors.

This analysis shows a robust prioritization with a notable focus on environmental sustainability, alongside important but lesser emphasis on social and economic factors, which could reflect the organization’s strategic objectives or sector-specific mandates.

2.5. Implementation in R



```r
#############################################
# D3M : Assignment 2
# Problem - 2 : Multi-Objective Line-Balancing
# Group C
#############################################

# Importing the library
library(AHPhybrid)

title <- "Multi Objective Line Balancing Problem"
Alternatives <- c("Setting1","Setting2","Setting3","Setting4","Setting5","Setting6")
Qualitative_Criteria <- c("economic","social","environmental")
Quantitative_criteria <- c()
Quantitative_crit_min_max <- c()

n_alt <- length(Alternatives)
n_crit <- length(Qualitative_Criteria)
n_crit_Qual <- length(Qualitative_Criteria)
n_crit_Quant <- length(Quantitative_criteria)

# 3 * 3 Comparison Matrix
Criteria_Comparison <- matrix(c(
  1,3,5,
  1/3,1,4,
  1/5,1/4,1),
  ncol = n_crit, nrow = n_crit, byrow = TRUE)

Alternatives_comparison_qualit_crit <- list(
  matrix(c(
    8, 8, 7, 7, 7, 7,
     8, 8, 7, 7, 7, 7,
     8, 1, 8, 8, 8, 7,
     8, 8, 8, 8, 8, 7,
     9, 9, 8, 8, 8, 8,
     9, 9, 8, 8, 8, 8), ncol = n_alt, nrow = n_alt, byrow = TRUE),

  matrix(c(
    4, 9, 4, 6, 6, 7,
    1, 4, 1, 2, 2, 3,
  4, 9, 4, 6, 6, 6,
```

```
# 3 * 3 Comparison Matrix
Criteria_Comparison <- matrix(c(
  1,3,5,
  1/3,1,4,
  1/5,1/4,1),
  ncol = n_crit, nrow = n_crit, byrow = TRUE)

Alternatives_comparison_qualit_crit <- list(
  matrix(c(
    8, 8, 7, 7, 7, 7,
     8, 8, 7, 7, 7, 7,
     8, 1, 8, 8, 8, 7,
     8, 8, 8, 8, 8, 7,
     9, 9, 8, 8, 8, 8,
     9, 9, 8, 8, 8, 8), ncol = n_alt, nrow = n_alt, byrow = TRUE),

  matrix(c(
    4, 9, 4, 6, 6, 7,
    1, 4, 1, 2, 2, 3,
  4, 9, 4, 6, 6, 6,
  6, 9, 3, 4, 4, 4,
  3, 7, 3, 5, 4, 5,
  2, 6, 2, 4, 4, 4 ), ncol = n_alt, nrow = n_alt, byrow = TRUE),


  matrix(c(
    4, 2, 2, 5, 6, 3,
    6, 4, 4, 7, 9, 4,
  6, 4, 4, 8, 9, 4,
  3, 2, 2, 4, 5, 2,
  2, 1, 1, 3, 4, 1,
  6, 4, 4, 7, 8, 4), ncol = n_alt, nrow = n_alt, byrow = TRUE)
  )

# Solving the Model
AHPhybrid(title, Alternatives, Qualitative_Criteria,
          Quantitative_criteria, Quantitative_crit_min_max,
          n_alt, n_crit, n_crit_Qual, n_crit_Quant,
          Criteria_Comparison, Alternatives_comparison_qualit_crit,
          Alternatives_quantitative_crit)
```

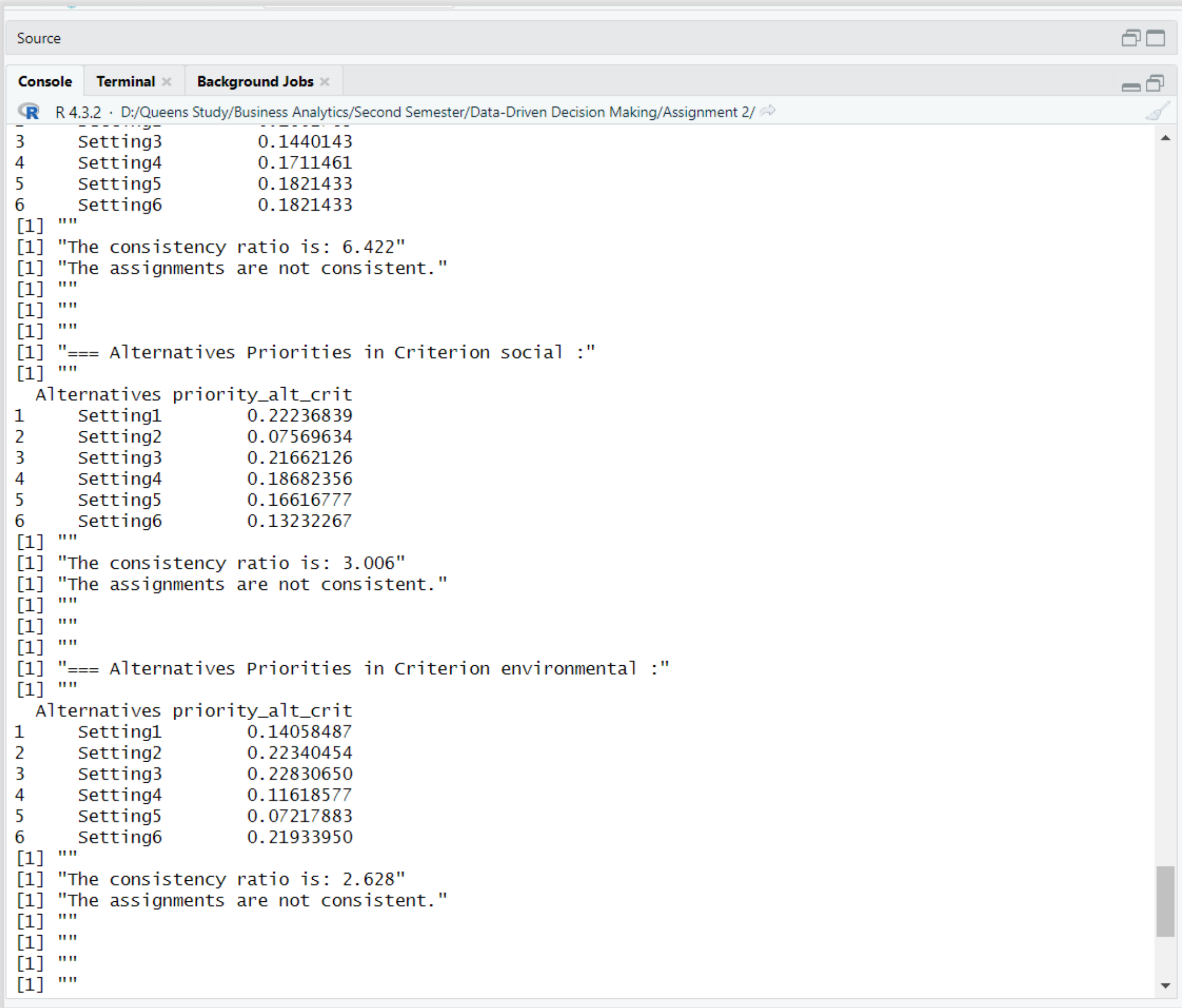
2.6. Output from R

Source

Console | Terminal | Background Jobs

R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/

```
> AHPhybrid(title, Alternatives, Qualitative_Criteria,
+           Quantitative_criteria, Quantitative_crit_min_max,
+           n_alt, n_crit, n_crit_Qual, n_crit_Quant,
+           Criteria_Comparison, Alternatives_comparison_qualit_crit,
+           Alternatives_quantitative_crit)
[1] "Multi Objective Line Balancing Problem"
[1] ""
[1] ""
[1] "===== Criteria Priorities:"
[1] ""
       criteria priority_crit
1      economic         0.619
2        social         0.284
3 environmental         0.096
[1] ""
[1] "The consistency ratio is: 0.075"
[1] ""
[1] "The assignments are consistent."
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] "=== Alternatives Priorities in Criterion economic :"
[1] ""
  Alternatives priority_alt_crit
1     Setting1         0.1602765
2     Setting2         0.1602765
3     Setting3         0.1440143
4     Setting4         0.1711461
5     Setting5         0.1821433
6     Setting6         0.1821433
[1] ""
[1] "The consistency ratio is: 6.422"
[1] "The assignments are not consistent."
[1] ""
[1] ""
[1] ""
[1] "=== Alternatives Priorities in Criterion social :"
[1] ""
```




```
3     Setting3         0.1440143
4     Setting4         0.1711461
5     Setting5         0.1821433
6     Setting6         0.1821433
[1] ""
[1] "The consistency ratio is: 6.422"
[1] "The assignments are not consistent."
[1] ""
[1] ""
[1] ""
[1] "=== Alternatives Priorities in Criterion social :"
[1] ""
  Alternatives priority_alt_crit
1     Setting1        0.22236839
2     Setting2        0.07569634
3     Setting3        0.21662126
4     Setting4        0.18682356
5     Setting5        0.16616777
6     Setting6        0.13232267
[1] ""
[1] "The consistency ratio is: 3.006"
[1] "The assignments are not consistent."
[1] ""
[1] ""
[1] ""
[1] "=== Alternatives Priorities in Criterion environmental :"
[1] ""
  Alternatives priority_alt_crit
1     Setting1        0.14058487
2     Setting2        0.22340454
3     Setting3        0.22830650
4     Setting4        0.11618577
5     Setting5        0.07217883
6     Setting6        0.21933950
[1] ""
[1] "The consistency ratio is: 2.628"
[1] "The assignments are not consistent."
[1] ""
[1] ""
[1] ""
[1] ""
```

```
Source
Console   Terminal   Background Jobs
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
[1] ""
[1] ""
[1] ""
[1] "=== Alternatives Priorities in Criterion environmental :"
[1] ""
  Alternatives priority_alt_crit
1     Setting1        0.14058487
2     Setting2        0.22340454
3     Setting3        0.22830650
4     Setting4        0.11618577
5     Setting5        0.07217883
6     Setting6        0.21933950
[1] ""
[1] "The consistency ratio is: 2.628"
[1] "The assignments are not consistent."
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] "=====Alternatives priorities for each criterion:"
          economic     social environmental
Setting1 0.1602765 0.22236839    0.14058487
Setting2 0.1602765 0.07569634    0.22340454
Setting3 0.1440143 0.21662126    0.22830650
Setting4 0.1711461 0.18682356    0.11618577
Setting5 0.1821433 0.16616777    0.07217883
Setting6 0.1821433 0.13232267    0.21933950
[1] ""
[1] ""
[1] ""
[1] ""
[1] "===== Global Index :"
[1] "Final Results"
[1] "Setting1 = 0.175"
[1] "Setting3 = 0.173"
[1] "Setting6 = 0.172"
[1] "Setting4 = 0.17"
[1] "Setting5 = 0.167"
[1] "Setting2 = 0.141"
>
```

3. Problem-3: Airline Catering

3.1. Decision variables and objective function

Let $x_1 \in \mathbb{R}^+$ and $x_2 \in \mathbb{R}^+$ be **number of kg** of special and basic fillings the caterer prepares, respectively. Therefore, the total revenue (in CHF) is

$$\begin{aligned} \text{revenue} &= (\text{weight of special filling}) \times \text{price per kg of special filling} \\ &+(\text{weight of basic filling}) \times \text{price per kg of basic filling} = x_1 p_1 + x_2 p_2, \end{aligned} \tag{1}$$

where p_1 and p_2 are the prices per kilogram of special and basic fillings, respectively, measured in **CHF/kg**.

From the question, the relationship between the demand and the price is given by

$$d_1 = 190 - 25p_1, \tag{2}$$

$$d_2 = 250 - 50p_2, \tag{3}$$

where d_1 and d_2 are the demands (**in kg**) of special and basic fillings, respectively.

By substituting (2) and (3) into (1), we obtain

$$\begin{aligned} \text{revenue} &= x_1\left(\frac{190-d_1}{25}\right)+x_2\left(\frac{250-d_2}{50}\right) \\ &= x_1\left(\frac{38}{5}-\frac{d_1}{25}\right)+x_2\left(5-\frac{d_2}{50}\right) \qquad (4) \end{aligned}$$

3.2. Constraints

Since the catering company has 20 kg of raspberries, and each kg of special consists of 0.2 kg raspberry, while each kg of basic consists of 0.2 kg raspberry, we have

$$0.2x_1 + 0.2x_2 \leq 20. \qquad (5)$$

Similarly, since the catering company has 60 kg of premium-quality chocolate, and each kg of special and basic consists of 0.8 kg and 0.3 kg of premium chocolate, respectively, we have

$$0.8x_1 + 0.3x_2 \leq 60. \qquad (6)$$

In addition, the special and basic fillings that the catering company produces cannot exceed the corresponding demands. Therefore, we have the following constraints

$$x_1 \leq d_1, \qquad (7)$$

$$x_2 \leq d_2. \qquad (8)$$

3.3. Final mathematical formulation

Our objective is to select x_1, x_2, d_1, d_2 that maximize the revenue given by (4), subject to the constraints (5)-(8). Mathematically, our optimization problem can be formulated as

$$\begin{aligned} &Max \quad \text{revenue} = x_1\left(\frac{38}{5}-\frac{d_1}{25}\right)+x_2\left(5-\frac{d_2}{50}\right) \\ &\text{s.t.} \quad 0.2x_1 + 0.2x_2 \leq 20, \\ &\qquad 0.8x_1 + 0.3x_2 \leq 60, \\ &\qquad x_1 \leq d_1, \\ &\qquad x_2 \leq d_2, \qquad \text{and } x_1, x_2 \in \mathbb{R}^+. \end{aligned}$$

Given x_1 and x_2, the objective function **"$\mathbf{revenue}$" decreases when d_1 and d_2 increase**. Thus, at the optimal point, the constraints $x_1 \leq d_1$ and $x_2 \leq d_2$ **must hold with equality**. As a result, the original optimization problem can be simplified as

$$\begin{aligned} &Max \quad \text{revenue} = x_1\left(\frac{38}{5}-\frac{x_1}{25}\right)+x_2\left(5-\frac{x_2}{50}\right) \\ &\text{s.t.} \quad 0.2x_1 + 0.2x_2 \leq 20, \\ &\qquad 0.8x_1 + 0.3x_2 \leq 60, \quad \text{and } x_1, x_2 \in \mathbb{R}^+. \end{aligned}$$

Note that the objective function includes x_1^2 and x_2^2 which are **nonlinear**. We, therefore, can use a **metaheuristic** (i.e., the local search & threshold accepting method) to solve the above optimization problem.

3.4. Graphical representation and components of metaheuristics

Two constraints in the optimization problem can be graphically illustrated in Fig. 2. A point (x_1, x_2) in the area with diagonal stripe pattern is the solution representation.

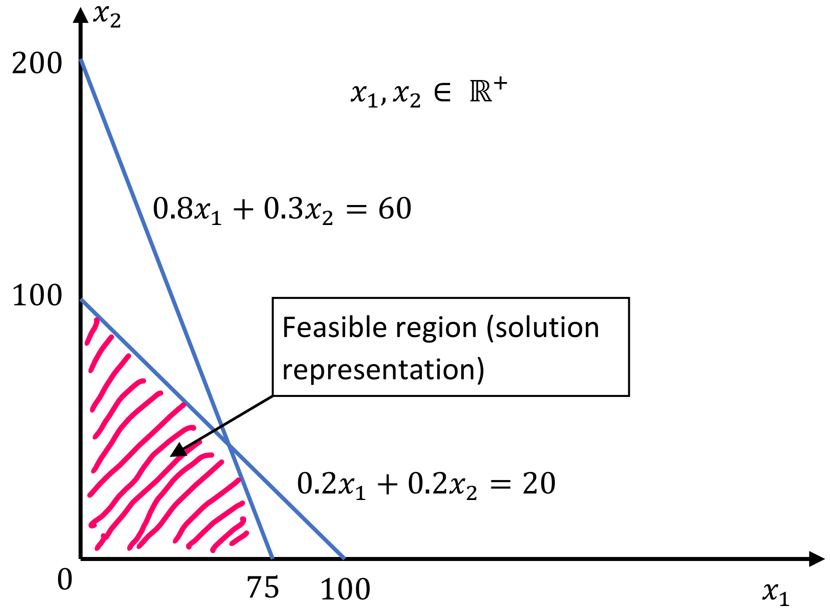


Fig. 2. Graphical illustration of the constraint.

Using the above representation, we construct components of the metaheuristic as follows:

- **Initial solution**: We keep randomly choose a point until it satisfies all constraints, i.e. this point is in the area with diagonal stripe pattern. Note that from Fig. 3.1, x_1 should be between 0 and $x_{1,\max} = 75$, while x_2 should be between 0 and $x_{2,\max} = 100$.
- Calculate the value of the **objective function value**, using
$$\text{revenue} = x_1\left(\frac{38}{5} - \frac{x_1}{25}\right) + x_2\left(5 - \frac{x_2}{50}\right)$$
- **Neighborhood search**: we use a window that spans 2 units around the current point to find the new solution. Note that we **choose a window of $2 \times 2 = 4$ units because the values of x_1 and x_2 are less than 100**. Then, we check whether this new point satisfies the constraints. If it does not, we discard it and search again until we find a satisfying point.

Remark: We use R to solve our optimization problem with meta-heuristic method. The LSopt and TAopt functions of the NMOF package in R are used. Since these functions are coded for minimization, we add the negative sign to our objective function. This comes from the fact that $Max\ f(x)$ is equivalent to $Min\ (-f(x))$. **We also set seed with an integer (1) to make it reproducible**.

3.5. Implementation using ‘NMOF’ package in R.

```r
### ===========================================
### Problem-3: Airline Catering
### x1 is the number of kg of special filling
### x2 is the number of kg of basic filling

### -------------------------------------------
### Import the NMOF package.

library(NMOF)

set.seed(1) # set seed for reproducible output

### input parameters of the problem

ns = 300L         # number of iterations
p1Const1 = 38/5   # special filling first price constant
p1Const2 = 1/25   # special filling second price constant
p2Const1 = 5      # basic filling first price constant
p2Const2 = 1/50   # basic filling second price constant

x1Max = 75        # maximum value of x1 in the feasible region
x2Max = 100       # maximum value of x2 in the feasible region
halfWindow = 2    # window used for neighborhood search

### Checking whether a given point is in the feasible region
FeasibilityCheck = function(x){
  IsFeasible = TRUE
  Violation = (x[1] < 0) | (x[2] < 0) |
    (0.2*x[1]+0.2*x[2] > 20) | (0.8*x[1]+0.3*x[2] > 60)
  if (Violation){
    IsFeasible = FALSE
  }
  IsFeasible
}

data = list(p1Const1 = p1Const1, p1Const2 = p1Const2,
            p2Const1 = p2Const1, p2Const2 = p2Const2,
            x1Max = x1Max, x2Max = x2Max,
            halfWindow = halfWindow)
```

```r
data = list(p1Const1 = p1Const1, p1Const2 = p1Const2,
            p2Const1 = p2Const1, p2Const2 = p2Const2,
            x1Max = x1Max, x2Max = x2Max,
            halfWindow = halfWindow)


### Initial solution
makeRandomSol = function(data) {
  x0 = c(1000,1000)
  IsFeasible = FeasibilityCheck(x0)
  while ( !IsFeasible ) {
    x0[1] = runif(1, min = 0, max = data$x1Max)
    x0[2] = runif(1, min = 0, max = data$x2Max)
    IsFeasible = FeasibilityCheck(x0)
  }
  x0
}
x0 = makeRandomSol(data)

### Neighborhood searching
neighbor = function(xc, data) {
  xn = xc + c(1000,1000)
  IsFeasible = FeasibilityCheck(xn)
  while ( !IsFeasible ) {
    xn[1] = xc[1] + runif(1, min = -data$halfWindow, max = data$halfWindow)
    xn[2] = xc[2] + runif(1, min = -data$halfWindow, max = data$halfWindow)
    IsFeasible = FeasibilityCheck(xn)
  }
  xn
}

### objective function (the revenue)
OF = function(xn, data){
  -( (data$p1Const1-data$p1Const2*xn[1])*xn[1] +
       (data$p2Const1-data$p2Const2*xn[2])*xn[2] )
}

algo = list(nS = ns, neighbour = neighbor, x0 = x0,
            printBar = TRUE, printDetail = TRUE)
```

```r
algo = list(nS = ns, neighbour = neighbor, x0 = x0,
            printBar = TRUE, printDetail = TRUE)

### (1) *Local Search*
sol1 = LSopt(OF, algo = algo, data = data)
x0
sol1$xbest
sol1$OFvalue
par(ylog = FALSE)
plot(sol1$Fmat[ ,2L],type = "l", #log = "y",
     ylim = c(min(pretty(sol1$Fmat[ ,2L])),
              max(pretty(sol1$Fmat[ ,2L]))),
     xlab = "iterations", ylab = "OF value", main = "Objective value versus number of iterations", col = "blue")

### (2) *Threshold Accepting*
algo$nT <- 10L
algo$nS <- ceiling(algo$nS/algo$nT)
sol2 <- TAopt(OF, algo = algo, data = data)
x0
sol2$xbest
sol2$OFvalue
lines(cummin(sol2$Fmat[ ,2L]),type = "l", lty = 2, col = "red")

### Best Quantity and Revenue
print(paste("Number of kg of special filling: ", sol2$xbest[1]))
print(paste("Number of kg of basic filling: ", sol2$xbest[2]))
print(paste("Best revenue in CHF: ", -sol2$OFvalue))

### Price for the two cake fillings
p1= p1Const1-sol2$xbest[1]*p1Const2
p2= p2Const1-sol2$xbest[2]*p2Const2
print(paste("Price for special cake filling in CHF: ", p1))
print(paste("Price for basic cake filling in CHF: ", p2))

### Remaining raspberry and premium chocolate
remain_ras = 20 - (sol2$xbest[1]*0.2 + sol2$xbest[2]*0.2)  #Raspberries
remain_choc = 60 - (sol2$xbest[1]*0.8 + sol2$xbest[2]*0.3) #premium chocolate
print(paste("Remaining raspberry in kg: ", remain_ras))
print(paste("Remaining premium chocolate in kg: ", remain_choc))
```

3.6. Output solution from R.

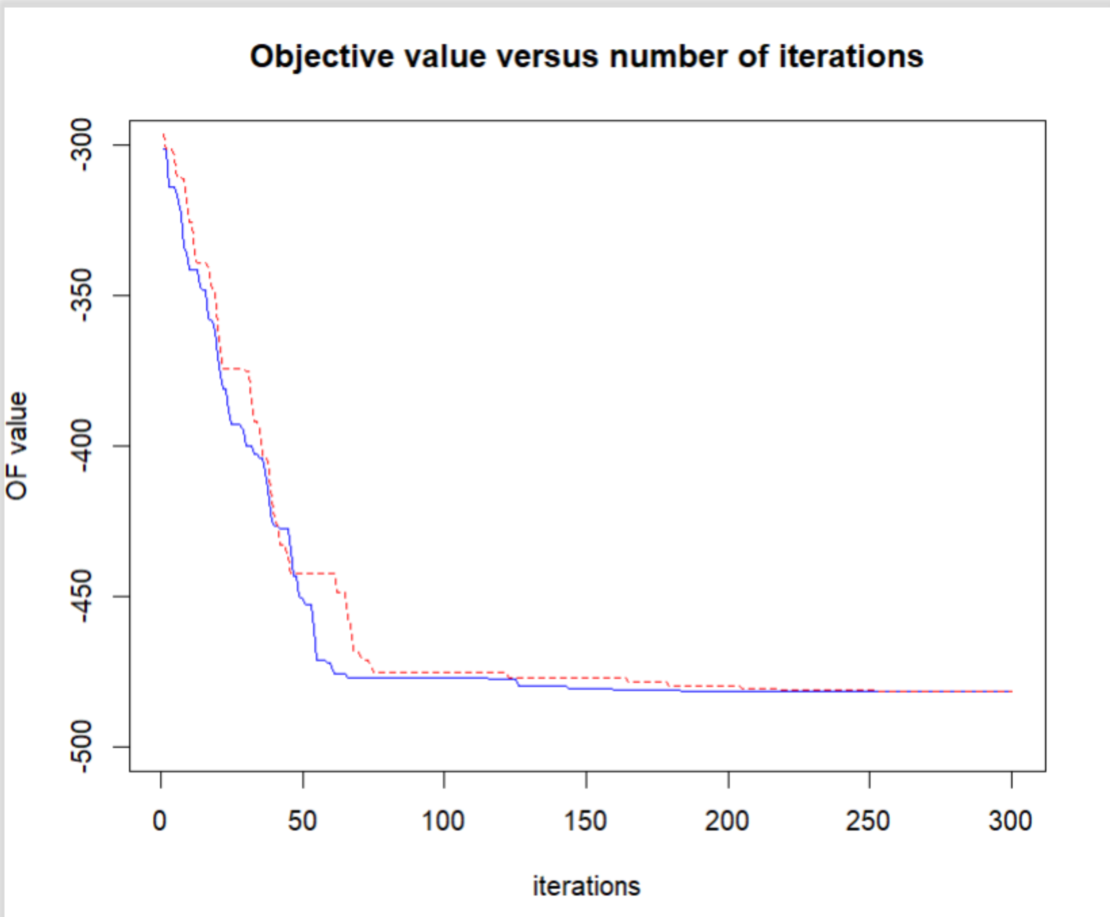


Figure 3. The objective function values versus the number of iterations. The red line represents values using threshold accepting, while the blue one represents the local search. As observed, revenue increase significantly after 50 iterations. After 200 iterations, the results of both methods are consistent.

```
Console   Terminal ×   Background Jobs ×
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/

Local Search.
Initial solution:  -293.8453
  |==============================================================================================| 100%
Finished.
Best solution overall: -481.3226
> x0
[1] 19.91315 37.21239
> sol1$xbest
[1] 53.93221 46.03329
> sol1$OFvalue
[1] -481.3226
> par(ylog = FALSE)
> plot(sol1$Fmat[ ,2L],type = "l", #log = "y",
+      ylim = c(min(pretty(sol1$Fmat[ ,2L])),
+               max(pretty(sol1$Fmat[ ,2L]))),
+      xlab = "iterations", ylab = "OF value", main = "Objective value versus number of iterations")
>
> ### (2) *Threshold Accepting*
> algo$nT <- 10L
> algo$nS <- ceiling(algo$nS/algo$nT)
> sol2 <- TAopt(OF, algo = algo, data = data)

Threshold Accepting

  Computing thresholds ...
  |==============================================================================| 100%
  OK
  Estimated remaining running time: 0.009 secs

  Running Threshold Accepting ...
  Initial solution: -293.8453
  |==============================================================================| 100%
  Finished.
  Best solution overall: -481.4549
> x0
[1] 19.91315 37.21239
> sol2$xbest
[1] 54.76958 45.21728
> sol2$OFvalue
[1] -481.4549
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.2 · D:/Queens Study/Business Analytics/Second Semester/Data-Driven Decision Making/Assignment 2/
  |==============================================================================| 100%
  OK
  Estimated remaining running time: 0.0105 secs

  Running Threshold Accepting ...
  Initial solution: -293.8453
  |==============================================================================| 100%
  Finished.
  Best solution overall: -481.4549
> x0
[1] 19.91315 37.21239
> sol2$xbest
[1] 54.76958 45.21728
> sol2$OFvalue
[1] -481.4549
> lines(cummin(sol2$Fmat[ ,2L]),type = "l", lty = 2)
>
> ### Best Quantity and Revenue
> print(paste("Number of kg of special filling: ", sol2$xbest[1]))
[1] "Number of kg of special filling:  54.7695808920544"
> print(paste("Number of kg of basic filling: ", sol2$xbest[2]))
[1] "Number of kg of basic filling:  45.2172841187567"
> print(paste("Best revenue in CHF: ", -sol2$OFvalue))
[1] "Best revenue in CHF:  481.454900068218"
>
> ### Price for the two cake fillings
> p1= p1Const1-sol2$xbest[1]*p1Const2
> p2= p2Const1-sol2$xbest[2]*p2Const2
> print(paste("Price for special cake filling in CHF: ", p1))
[1] "Price for special cake filling in CHF:  5.40921676431782"
> print(paste("Price for basic cake filling in CHF: ", p2))
[1] "Price for basic cake filling in CHF:  4.09565431762487"
>
> ### Remaining raspberry and premium chocolate
> remain_ras = 20 - (sol2$xbest[1]*0.2 + sol2$xbest[2]*0.2)  #Raspberries
> remain_choc = 60 - (sol2$xbest[1]*0.8 + sol2$xbest[2]*0.3) #premium chocolate
> print(paste("Remaining raspberry in kg: ", remain_ras))
[1] "Remaining raspberry in kg:  0.00262699783779397"
> print(paste("Remaining premium chocolate in kg: ", remain_choc))
[1] "Remaining premium chocolate in kg:  2.61915005072952"
>
```

3.7. Decision

About 54.77 kg special cake filling and 45.22 kg basic cake filling are made for the total revenue of around CHF481.45, with the prices for special and basic fillings of CHF5.41 and CHF4.10, respectively. In addition, approximately 0.0026 kg raspberries and 2.62 kg premium chocolate are left over.

Meeting minutes and equally declaration

Meeting 1: April 8th, 2024 (All members present)

The primary objective of the initial meeting was to foster acquaintance among team members, identifying and outlining the activities essential for accomplishing the assignment, as well as orchestrating the distribution of tasks among team members.

Meeting 2: April 10th, 2024 (All members present)

Problem two was thoroughly analysed and discussed by the group, culminating in the formulation of strategies to effectively address the challenges posed by the MOLB game.

Meeting 3: April 18th, 2024 (All members present)

Discussion regarding all three problems was undertaken, with team members engaging in a comprehensive analysis. Following this, proofreading was conducted collectively, ensuring meticulous scrutiny for any errors or inconsistencies. Appropriate modifications were made wherever deemed necessary, enhancing the overall quality of the report. Subsequently, discussions ensued concerning the finalization of the report, wherein consensus was sought on its content and presentation.

Tasks executed by team members:

All the team members worked together on problem 1 and also all team members have equal contribution towards report

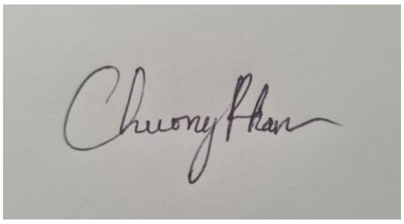
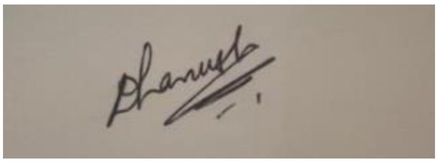
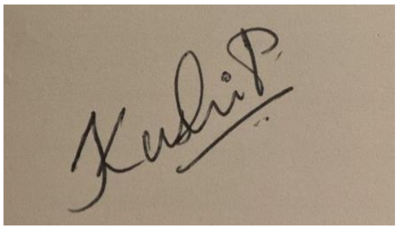
- **Chuong Pham -**
  Problem 3
  Setting 3 of problem 2.
  Problem 1 and Report Writing (Contributed by all members)
- **Dhanush** -
  Setting 2 of problem 2

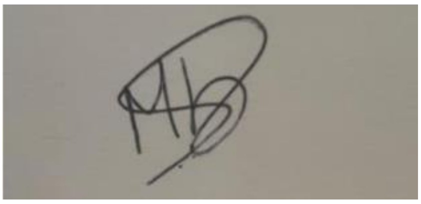
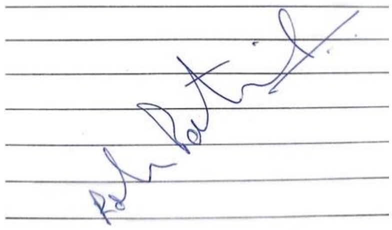
Problem 1 and Report Writing (Contributed by all members)

- **Kushi -**
  Setting 4 of problem 2
  Problem 1 and Report Writing (Contributed by all members)
- **Manogna -**
  Setting 5 of problem 2
  Problem 1 and Report Writing (Contributed by all members)
- **Rohan -**
  Setting 6 of problem 2
  Problem 1 and Report Writing (Contributed by all members)
- **Thomas -**
  Setting 1 of problem 2
  Problem 1 and Report Writing (Contributed by all members)

Signed Declaration:

We all hereby declare that we have coordinated and completed the group assignment for **Data Driven Decision Making** submitted to Queens' University Belfast, the tasks that were. completed and submitted are as mentioned in the activity report above.

Name	Signature
Chuong Pham - 40411407	
Dhanush Mathighatta Shobhan Babu – 40412492	
Kushi Paramesh Kogilvadi – 40411893	

Manogna Bidarahalli Raghothamachar - 40426970	
Rohan Mahesh Patil - 40395741	
Thomas Mark Agnew - 40316391	

References

Ahmadi, T. and van der Rhee, B., 2023. Multiobjective Line Balancing Game: Collaboration and Peer Evaluation. INFORMS Transactions on Education, 23(3), pp.179-195.

Groover MP (2016) Automation, Production Systems, and Computer Integrated Manufacturing (Pearson Education India).