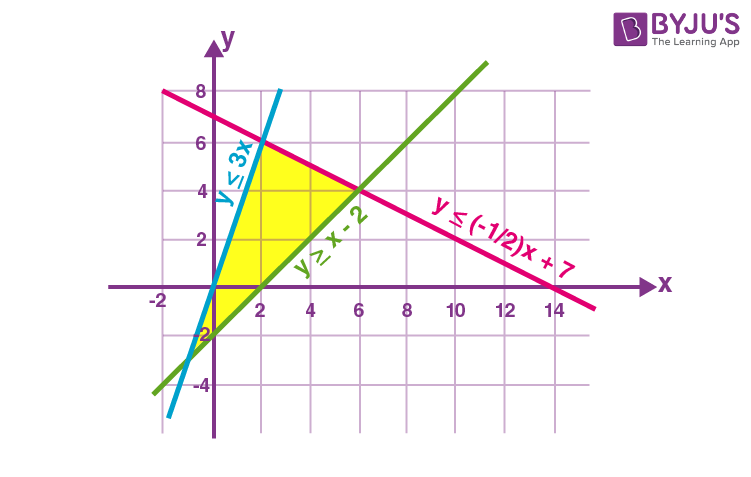
**AM 3038 / AM 3084/ FM 3036**

**Continuous Assessment 01**

Report 1

Sahashra Dinasiri (S14980)

Udani Anupama (S15048)

Lasal Sandeepa Hettiarachchi (S14815)

Department of Mathematics

University of Colombo

Sri Lanka

November 23 2022

Table of Contents

[Problem Description 3](#_Toc120097694)

[Assumptions 4](#_Toc120097695)

[Model Development 5](#_Toc120097696)

[Decision variables 5](#_Toc120097697)

[Objective function 5](#_Toc120097698)

[Constraints 5](#_Toc120097699)

[Path to the Solution 6](#_Toc120097710)

[Results 7](#_Toc120097711)

[Post optimal analysis 8](#_Toc120097715)

[Bibliography 9](#_Toc120097716)

# Problem Description

According to the problem given we have to find the best way for the TelZee to give the four promotions to the customers. To obtain the best way we have to make the total profit to be maximum. Assuming that there is no any cost for the data which is not considered as an extra data we can consider the total revenue of the past month as the profit gained from those used data. Since there are promotional data packages, the extra data will give a revenue of Rs. 250.00 per extra GB. But the company have to pay Rs. 150.00 for each extra GB. Therefore, the profit will be Rs. 100.00 per extra GB. Now we have the total profit as the sum of profit earned from the used data in past month and profit earned from extra GBs. Therefore, we have to maximize that total profit and it will be the objective function of the model.

Let us find how to allocate extra GB for each data package in order to maximize the profit. Therefore, we define decision variables as continuous variables where x\_i are the number of extra GBs that can be allocated for ith data package.

In the problem we are given that the capitalized total amount for all promotions is Rs. 8000.00. Therefore, we have to limit the number of extra GBs. From that limitation we can make a constraint to the model. Assuming that one customer can use at most one package and at most one extra GB we can conclude that the total number of extra GBs cannot exceed the limit of 83. That will give us another constraint to the model.

From above description we can build up the model. Then using some python implementation, we will be able to solve the model and find the optimum.

The problem can have some alternative models as well. If we define decision variables as Boolean variables where, xi j=The Boolean decision variable of the ith customer using the jth package. But if we define our decision variable as above the model that will cause to a naive solution which will give a high time complexity.

# Assumptions

* One Customer can use at most one package.
* One Customer can use at most one extra GB.
* All customers use the relevant promotions and use the extra data this month.
* Assume there is no cost for the data that is not under the promotional extra data for TelZee
* For a given package, 1 Extra GB is offered only if the customer has exceeded their last month’s usage

# Model Development

## Decision variables

## Objective function

## Constraints

## (One customer can use at most one extra GB)

If a single customer can subscribe to one package that means that all the customers together should have less than 83 GB of extra data

## (The company is willing to spend a capital cost of 8000 for promotion)

## 

--------------------------------------------------------------------------------------------------------------

## Alternative model

## Decision variables

## Objective function

## 

* Total data usage by all 83 customers during the given month = 867 units
* Revenue from each GB a customer use = Rs.250.00
* Cost of a promotional 1 GB to the company = Rs.150.00
* Profit from each promotional 1GB = Rs. (250.00 - 150.00)

Constraints

* : for

## 

# Path to the Solution

from pulp import LpMaximize, LpProblem, LpStatus, lpSum, LpVariable

total\_revenue\_from\_last\_month = 216750

# Create the model

model = LpProblem(name="Telzee-problem", sense=LpMaximize)

# Initialize the decision variables

x1 = LpVariable(name="x1", lowBound=0)

x2 = LpVariable(name="x2", lowBound=0)

x3 = LpVariable(name="x3", lowBound=0)

x4 = LpVariable(name="x4", lowBound=0)

# Add the constraints to the model

model += (x1 + x2 + x3 + x4 <= 83, "One customer can use atmost one extra GB")

model += (150 \* x1 + 150 \* x2 + 150 \* x3 + 150 \* x4 <= 8000, "The company is willing to spend a capital cost of 8000 for promotion")

# #Objective function and adding to the model

obj\_func = (250 - 150) \* (x1 + x2 + x3 + x4) + total\_revenue\_from\_last\_month

model += obj\_func

print(model)

print()

print()

print("----------------------------------------------------------------")

print()

print()

# Solve the problem

status = model.solve()

print(f"status: {model.status}, {LpStatus[model.status]}")

print(f"objective: {model.objective.value()}")

for var in model.variables():

print(f"{var.name}: {var.value()}")

The path for the solution was implemented using a python code that has been mentioned above. The solution we received according to the above code is as below in the results section.

Code of the alternative solution

from pulp import LpMaximize, LpProblem, LpStatus, lpSum, LpVariable

import numpy

# Create the model

model = LpProblem(name="Telzee-problem2", sense=LpMaximize)

matrix = [[[LpVariable(name="x"+str(j)+str(j), lowBound=0)] for j in range(5)] for i in range(84)]

print(matrix)

sigma = 0

for i in range(0,84):

sigma = matrix[i][0] + matrix[i][1] + matrix[i][2] + matrix[i][3]

# # Add the constraints to the model

# for i in range(1,84):

# model += (x1 + x2 + x3 + x4 <= 83, "One customer can use atmost one extra GB")

# model += (150 \* x1 + 150 \* x2 + 150 \* x3 + 150 \* x4 <= 8000, "The company is willing to spend a capital cost of 8000 for promotion")

#

# Add the constraints to the model

for i in range(1,84):

model += (matrix[i][0] + matrix[i][1] + matrix[i][2] + matrix[i][3] <= 1, "One customer can use atmost one extra GB"+str(i))

model += (150 \* sigma <= 8000, "x")

# #Objective function and adding to the model

obj\_func = (250 \* 867) + (250-150) \* sigma

model += obj\_func

print(model)

print()

print()

print("----------------------------------------------------------------")

print()

print()

# Solve the problem

status = model.solve()

print(f"status: {model.status}, {LpStatus[model.status]}")

print(f"objective: {model.objective.value()}")

for var in model.variables():

print(f"{var.name}: {var.value()}")

# Results

Text

Description automatically generated

Figure 1: Screenshot of the complete model

Text

Description automatically generated

Figure 2: Output of the implemented python code with the solution.

From the above results the solution that was obtained was that for the company to allocate 53.3333 data for the Drowzee package and 0 data for all the other packages. But this Problem which is a continues problem. The answer is wrong. This is due to the constraints of the above model being error prone. The model does not properly match that the customer can at most have one package. The constraint

## (One customer can use at most one extra GB)

Doesn’t optimally model it. Also, the above model does not properly capture the modeling that certain packages offer extra GB when the pervious quota is over. Therefore, the answer is dictated by the constraint

## (The company is willing to spend a capital cost of 8000 for promotion)

## (The company is willing to spend a capital cost of 8000 for promotion)

Therefore, an alternative model to capture the above errors with binary decision variables were created. The error with this approach is that 83 \* 4 decision variables are defined and to solve that using a naïve approach the time complexity is 2^n which is extremely high.

# Post optimal analysis

When we consider the post optimal analysis, there are two major sections.

1. Changes affecting feasibility
2. Changes affecting optimality

Changes of right-hand side of the constraint and adding a new constraint make changes that affect feasibility. On the other hand, the changes in original objective coefficient and addition of a new variable make changes that affect optimality.

# Bibliography

Stojiljković, M., n.d. *Hands-On Linear Programming: Optimization With Python – Real Python.* [Online]   
Available at: https://realpython.com/linear-programming-python/#linear-programming-explanation  
[Accessed 23 11 2022].