

**Data-Driven Decision-Making**

**MGT7180**

**2020-2021**

Assignment 1

Vasileios Gounaris Bampaletsos

40314803

Contents

[Contents 2](#_Toc66795872)

[Problem 1 3](#_Toc66795873)

[Problem 1a 3](#_Toc66795874)

[Problem-1b 9](#_Toc66795875)

[Problem 2 14](#_Toc66795876)

[Problem 3 20](#_Toc66795877)

[Problem 4 28](#_Toc66795878)

# Problem 1

Supermarket Shelf Space

A supermarket wants to know some things about their own brand of peanut butter among other brands.

The profit for their own brand of peanut butter is 0.35 euros per jar and 0.25 euros for all other brands.

Also they have the information that the shelf space allocated to peanut butter is 0.5 and each jar of any brand of peanut butter takes 50.

## Problem 1a

In the first question the supermarket’s operations management wants to know how many jars of their own brand and how many jars of other brands combined to stock each day on the allocated shelf space for maximum profit.

Also we have the information from the daily sales records that the number of jars sold of their own brand NEVER exceeds half as many jars of the other brands combine.

First job is to split and laybel the two products.

|  |  |
| --- | --- |
| Product 1 – | Supermarket’s brand(jars) |
| Product 2 - | Other brands(jars) |

Second job is to transform the numbers for space in the same scale. For this task I select to trasform the .

I select this transformation because we have already that the shelf space for a jar is in cm and the transformation from cm to m have a lot of digits and it is a very small number. The transformation from m to cm is more clear for our purpose.

shelf space each jar =

shelf space for all jars =

The objective in task a is the number of jars (own & other) are needed to have the maximum profit. In other words the manager wants to know how many jars should be in the shelf space for maximum profit.

**Build the equations**

Z: objective value(profit in euros) to be maximized

)

Our goal is to make an equation that calculates the profit. For this reason we define the equation Z. The coefficients of the variables x are the values of each product respectively. The product of the supermarket costs 0.35 euros and that is why it is multiplied by the number of jars that the company will sell from its own product. Respectively, the product of other brands costs 0.25 euros per jar and is multiplied by the variable that indicates the number of jars of other brands that will be sold.

The objective function should find the maximum profit.

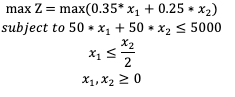
objective:

Then we have to add the constraints that refer to the problem. The first restriction is the available space that the supermarket has for the sale of peanut butter. The total space available to the supermarket for all peanut butter jars is 5000cm and each jar takes up 50cm.

The next constraint is for sales. The problem tell that the daily sales records indicate that the number of jars sold of the supermarket’s brand never exceeds half as many jars of the other brand combined. For that reason x1 never sold more than x2/2.

At the end should set the bounds for x1 and x2. They are count jars and for that reason should be greater than zero because cannot be negative.

At this point, it is good to add the whole problem in ompr and ipsolve and represent the entire problem together.

R input using ompr:

#===============================

#problem 1A

#In this question we need to find

#the number of the jars of their own brand

#and of the other brands on the allocated

#shelf space for MAXIMUM profit.

#===============================

#OMPR MODEL

#import the necesary packages for ompr

library(dplyr)

library(ROI)

library(ROI.plugin.glpk)

library(ompr)

library(ompr.roi)

#================================

#build the model with the objective equation

#and with the constraints

MIPModel() %>%

#================================

#decision variables and set the type as integer

#because we have jars(subjects)

add\_variable(x[i], i = 1:2, type = "integer") %>%

#==================================

#objective function

#x1 = jars (own brand), x2 = jars (other brands)

set\_objective(0.35\*x[1] + 0.25\*x[2], "max") %>%

#====================================

#constrains

#space per jar is 50cm2 and the max space is 5000cm2

#own brand"s sales are always half of others brand's sales

add\_constraint(50\*x[1] + 50\*x[2] <= 5000) %>% #shelf space

add\_constraint(x[1] <= x[2]/2) %>% #sales

set\_bounds(x[i], lb=0, i=1:2) %>% #set bounds for x1,x2

#======================================

#solving the problem and extract the results

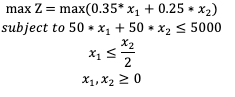
solve\_model(with\_ROI(solver="glpk", verbose=TRUE)) %>%

get\_solution(x[i]) %>%

filter(value > 0)

to get the solver message we give it TRUE.

objective value

R output ompr:

<SOLVER MSG> ----

GLPK Simplex Optimizer, v4.65

2 rows, 2 columns, 4 non-zeros

\* 0: obj = -0.000000000e+00 inf = 0.000e+00 (2)

\* 2: obj = 2.833333333e+01 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

GLPK Integer Optimizer, v4.65

2 rows, 2 columns, 4 non-zeros

2 integer variables, none of which are binary

Integer optimization begins...

Long-step dual simplex will be used

+ 2: mip = not found yet <= +inf (1; 0)

Solution found by heuristic: 28.3

+ 2: mip = 2.830000000e+01 <= tree is empty 0.0% (0; 1)

INTEGER OPTIMAL SOLUTION FOUND

<!SOLVER MSG> ----

The maximum daily profit for the whole peaut butter shelf space is 28.30 euros/day

variable i value

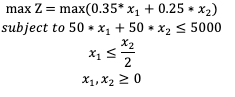
1 x 1 33

2 x 2 67

x1 = own brand

x2 = other brands

the supermarket should have 33 jars of their own brand and 67 jars of other brands to have maximum profit

R input using IpSolve:

#===================================

#IPSOLVE

#Problem-1a

#This script solves the problem for the supermarket

#shelf space with the ipsolve

#==============================

#import the ipsolve package

library(lpSolve)

#==============================

#create the objective function

#build a vector of coefficients

#from the objective function

objective <- c(

0.35, #profit per jar for own brand (x1)

0.25) #profit per jar for other brands (x2)

#==============================

#the constraints

#matrix with coefficients of constrains equations

const.mat <- matrix(c(

50, 50, #shelf space equation

2, -1, #the difference in sales

1, 0, #nonnegativity constraint for x1

0, 1), nrow=4, byrow = TRUE) #nonnegativity constraint for x2

#==============================

#the equality/inequality signs

const.dir <- c(

"<=", #inequality sign of the 1st constraint(shelf space)

"<=", #inequality sign of the 2nd constraint(sales' differnece)

">=", #inequaluty sign of the 3rd constraint(for x1)

">=") #inequaluty sign of the 4th constraint(for x2)

#==============================

#the right hand side(rhs) parameters(constants)

const.rhs <- c(

5000, #rhs value of the 1st constraint

0, #rhs value of the 2nd constraint

0, #rhs value of the 3rd constraint

0) #rhs value of the 4th constraint

#================================

#mathematical programming setting

#find the opt objective(MAX profit)

lp(direction="max", objective, const.mat, const.dir,

const.rhs, all.int = TRUE)

#find the opt solution(Values of x1 and x2)

lp(direction="max", objective, const.mat,

const.dir, const.rhs, all.int = TRUE)$solution

R Output using IpSolve:

Success: the objective function is 28.3

[1] 33 67

same result as the ompr output

The solution in problem, for maximum profit, is that the supermarket’s brand should has 33 jars and other brands combined should have 67 jars. Both models (ompr and ipsolve) find the same solution and the maximum profit is 28.30 euros/day.

## Problem-1b

For this problem the peanut butter from supermarket’s own brand is on discount. The sales records show that their own brand sells at least 1.2 times as many jars as other brands combined. The profit of their own brand decreased from 0.35 to 0.20 euros and the other brands remain steady in 0,25 euros.

The questions is the same as before, how many jars of each brand should be placed in the peanut butter shelf space for maximum profit, when the supermarket’s brand is on discount.

Z: objective value(profit in euros) to be maximized

)

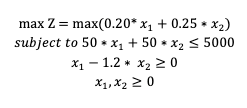
For that reason the objective equation should change because the price per jar for their own brand is now 0.20 euros and no 0.35.

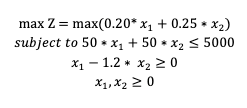
The constraint equation for the shelf space remains the same as before because the shelf space did not change.

The 2nd constraint for sales should change because now their own product is on discount and sell 1.2 times more than the other brands. For that reason:

Transport the 1.2x2 from left side to the right side and it takes a minus sign.

At the end should set the bounds for x1 and x2. They are count jars and for that reason should be greater than zero.



R input using ompr:

#====================================

#problem 1B

#In this question we need to find

#the number of the jars of their own brand

#and of the other brands on the allocated

#shelf space for MAXIMUM profit.

#BUT the supermarket makes a discount

#in their own brand jars and the sales and profit change

#======================================

#OMPR MODEL

#import the necesary packages for ompr

library(dplyr)

library(ROI)

library(ROI.plugin.glpk)

library(ompr)

library(ompr.roi)

#====================================

#build the model with the objective equation

#and with the constraints

MIPModel() %>%

#====================================

#decision variables and set the type as integer

#because we have jars(subjects)

add\_variable(x[i], i = 1:2, type = "integer") %>%

#====================================

#objective function to find the max profit

#x1 = jars (own brand), x2 = jars (other brands)

set\_objective(0.20\*x[1] + 0.25\*x[2], "max") %>%

#====================================

#constrains

#space per jar is 50cm2 and the max space is 5000cm2

#own brand"s sales are 1.2 times more than the other brands

add\_constraint(50\*x[1] + 50\*x[2] <= 5000) %>% #shelf space

add\_constraint(x[1] >= 1.2\*x[2]) %>% #sales

set\_bounds(x[i], lb=0, i=1:2) %>% #set bounds fro x1,x2

#======================================

#solving the problem and extract the results

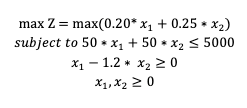
solve\_model(with\_ROI(solver="glpk", verbose=TRUE)) %>%

get\_solution(x[i]) %>%

filter(value > 0)

to get the solver message we give it TRUE.

objective value

R output using ompr:

<SOLVER MSG> ----

GLPK Simplex Optimizer, v4.65

2 rows, 2 columns, 4 non-zeros

\* 0: obj = -0.000000000e+00 inf = 0.000e+00 (2)

\* 2: obj = 2.227272727e+01 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

GLPK Integer Optimizer, v4.65

2 rows, 2 columns, 4 non-zeros

2 integer variables, none of which are binary

Integer optimization begins...

Long-step dual simplex will be used

+ 2: mip = not found yet <= +inf (1; 0)

Solution found by heuristic: 22.25

+ 2: mip = 2.225000000e+01 <= tree is empty 0.0% (0; 1)

INTEGER OPTIMAL SOLUTION FOUND

<!SOLVER MSG> ----

variable i value

1 x 1 55

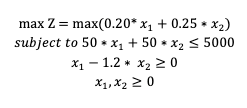
2 x 2 45

x1 = own brand

x2 = other brands

the supermarket should have 55 jars of their own brand and 45 jars of other brands to have maximum profit

The maximum daily profit for the whole peaut butter shelf space is 22.25 euros/day

R input using IpSolve:

#===================================

#IPSOLVE

#Problem-1b

#This script solves the problem for the supermarket

#shelf space with the ipsolve

#==============================

#import the ipsolve package

library(lpSolve)

#==============================

#create the objective function

#build a vector of coefficients

#from the objective function

objective <- c(

0.20, #profit per jar for own brand (x1)

0.25) #profit per jar for other brands (x2)

#==============================

#the constraints

#matrix with coefficients of constrains equations

const.mat <- matrix(c(

50, 50, #shelf space equation

1, -1.2, #the difference in sales

1, 0, #nonnegativity constraint for x1

0, 1), nrow=4, byrow = TRUE) #nonnegativity constraint for x2

#==============================

#the equality/inequality signs

const.dir <- c(

"<=", #inequality sign of the 1st constraint(shelf space)

">=", #inequality sign of the 2nd constraint(sales' differnece)

">=", #inequaluty sign of the 3rd constraint(for x1)

">=") #inequaluty sign of the 4th constraint(for x2

#=============================

#the right hand side(rhs) parameters(constants)

const.rhs <- c(

5000, #rhs value of the 1st constraint

0, #rhs value of the 2nd constraint

0, #rhs value of the 3rd constraint

0) #rhs value of the 4th constraint

#================================

#mathematical programming setting

#find the opt objective(MAX profit)

lp(direction="max", objective, const.mat, const.dir,

const.rhs, all.int = TRUE)

#find the opt solution(Values of x1 and x2)

lp(direction="max", objective, const.mat,

const.dir, const.rhs, all.int = TRUE)$solution

R output using IpSolve:

Success: the objective function is 22.25

[1] 55 45

same result as the ompr output

The solution in problem, for maximum profit, is that the supermarket’s brand should has 55 jars and other brands combined should have 45 jars. Both models (ompr and ipsolve) find the same solution and the maximum profit is 22.25 euros/day.

# Problem 2

Fleet Planning

In this problem, there is a company, NederVerzend, which is planning to buy some ships in three travelling range categories: long, medium and short.

The prices per ship for these categories are:

* long range ship: 70 million pounds
* medium range ship: 55 million pounds
* short range ship: 40 million pounds

The company has a spending limit in 2 billion pounds.

The net profit for each ship is:

* long range ship: 4.2 million pounds
* medium range ship: 3.1 million pounds
* short range ship: 2.6 million pounds

The company has another limit, it has enough personnel to manage 40 ships. At the same time the overall facilities capasity is equivalent to 60 short-range ships, where:

* medium-range = 1.5 short range
* long-range = 2 short-range

The CEO of the company wants to know how many of each category to buy for maximum profit and the amount of that profit.

The first job is to split and label each ship category.

|  |  |
| --- | --- |
| Long-range ship | X1 |
| Medium-range ship | X2 |
| Short-range ship | X3 |

Second job is to transform the pounds in the same dimension. For that reason, it is good to transform the billions pounds to million pounds.

I select the millions because they are more acceptable and more convinient for this problem, otherwise we will have demical data in our presentation.

Spending limit: 2 billion pounds = 2000 million pounds.

The objective in the task is the number of ships(long,medium,short) are needed to have the maximum profit. In other words the CEO wants to know how many ships should buy for maximum profit.

Build the equations

Z: objective value(profit in pounds) to be maximized

The goal is to find how many ships from each category, the company should buy to has maximum profit and the amount of that profit. For that reason our objective equation (Z) should be:

Then we have to add the constraints that refer to the problem. The first restriction is about the cost of the ships. The company has a limit in spending in 2000 million pounds.

The next constraint is about the personnel. The company has personnel which it can run 40 ships.

At the end, the company has limit in its facilities capacity. The company can have up to 60 short-range ships.

|  |  |
| --- | --- |
| Long-range ship |  |
| Medium-range ship |  |
| Short-range ship |  |

The final model is:

R input using ompr:

#================================================

#PROBLEM 2

#This script solves the LP model for the

#fleet planning

#In this model we need to find how many ships

#of each category should buy the company

#for maximum profit and find this number

#===============================================

#OMPR MODEL

#import the necesary packages for ompr

library(dplyr)

library(ROI)

library(ROI.plugin.glpk)

library(ompr)

library(ompr.roi)

#===============================================

#build the model with the objective equation

#and with the constraints

MIPModel() %>%

#================================================

#decision variables and set the type as integer

#because we have SHIPS(subjects)

add\_variable(x[i], i = 1:3, type = "integer") %>%

#================================================

#objective function to find the max profit

#x1 = long distance ship

#x2 = medium distance ship

#x3 = short distance ship

set\_objective(4.2\*x[1] + 3.1\*x[2] + 2.6\*x[3], "max") %>%

#================================================

#constrains

#set the price of each category and the budget of the company

#(in millions of dollas)

add\_constraint(70\*x[1] + 55\*x[2] + 40\*x[3] <= 2000) %>%

#set constrain for the maximum number of ships

#which they can manage

add\_constraint(x[1] + x[2] + x[3] <= 40) %>%

#set the limit of the facilities capasity

#company can manage up to 60 short ships

#long=2 shorts

#medium= 1.5 shorts

add\_constraint(x[1] <= 30) %>%

add\_constraint(x[2] <= 40) %>%

add\_constraint(x[3] <= 60) %>%

set\_bounds(x[i], lb=0, i=1:3) %>%

#================================================

#solving the problem and extract the results

solve\_model(with\_ROI(solver="glpk", verbose=TRUE)) %>%

get\_solution(x[i]) %>%

filter(value > 0)

to get the solver message we give it TRUE.

objective value

R output using ompr:

<SOLVER MSG> ----

GLPK Simplex Optimizer, v4.65

5 rows, 3 columns, 9 non-zeros

\* 0: obj = -0.000000000e+00 inf = 0.000e+00 (3)

\* 2: obj = 1.253333333e+02 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

GLPK Integer Optimizer, v4.65

5 rows, 3 columns, 9 non-zeros

3 integer variables, none of which are binary

Integer optimization begins...

Long-step dual simplex will be used

+ 2: mip = not found yet <= +inf (1; 0)

Solution found by heuristic: 124.8

+ 3: mip = 1.248000000e+02 <= tree is empty 0.0% (0; 3)

INTEGER OPTIMAL SOLUTION FOUND

<!SOLVER MSG> ----

variable i value

The maximum profit from ships will be 124.8 million pounds.

1 x 1 13

2 x 3 27

x1 = long-range ship

x2 = medium-ramge ship

x3 = short-range ship

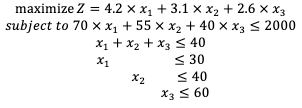
the company should buy 13 long-range ships and 27 short-range ships to achieve the maximum profit.

R input using IpSolve:

#===============================

#Problem-2

#This script solves the problem for fleet planning

#with the ipsolve in r

#==============================

#import the ipsolve package

library(lpSolve)

#==============================

#create the objective function

#build a vector of coefficients

#from the objective function

objective <- c(

4.2, #net profit per long ship

3.1, #net profit per medium ship

2.6) #net profit per small ship

#==============================

#the constraints

#matrix with coefficients of constrains equations

const\_mat <- matrix(c(

70, 55, 40, #1st constraint: the prices of each ship category(in millions)

1, 1, 1, #2nd constraint: the limitations for personnel

1, 0, 0, #3rd constraint: the limitation for long ships

0, 1, 0, #4th constraint: the limitation for medium ships

0, 0, 1), #5th constraint: the limitation for short ships

nrow=5,byrow=TRUE)

#==============================

#the equality/inequality signs

const\_dir<-c(

"<=", #inequality sign of the 1st constraint(prices)

"<=", #inequality sign of the 1st constraint(company's personnel)

"<=", #inequality sign of the 1st constraint(long ships)

"<=", #inequality sign of the 1st constraint(medium ships)

"<=") #inequality sign of the 1st constraint(short ships)

#==============================

#the right hand side(rhs) parameters(constants)

const\_rhs <- c(

2000, #rhs value of the 1st constraint (in millions)

40, #rhs value of the 2nd constraint (in ships)

30, #rhs value of the 3rd constraint (in ships)

40, #rhs value of the 4th constraint (in ships)

60) #rhs value of the 5th constraint (in ships)

#================================

#mathematical programming setting

#find the opt objective(MAX profit)

lp(direction="max", objective, const\_mat, const\_dir,

const\_rhs, all.int = TRUE)

#find the opt solution(Values of x1 and x2 and x3)

lp(direction ="max", objective, const\_mat, const\_dir,

const\_rhs, all.int = TRUE)$solution

R output using IpSolve:

Success: the objective function is 124.8

[1] 13 0 27

same result as the ompr output

The solution in problem, for maximum profit, is that the company should buy 13 long-range ships and 27 short-range ships. Both models (ompr and ipsolve) find the same solution and the maximum total profit for the company will be 124.8 million pounds.

# Problem 3

Production and Distibution

In this problem the main task is that the COO wants to know which plant should produce the supplement in that month and for which customers. First job is to identify the variables.

,

First job is to create the matrix of the distances between the cities. Its name is C\_dist

After create a matrix with order amount in kg per customer per plant. Its name is Mord\_am. This matrix is created by the replication (8times) of the vector of orders’ amount per customer(Vord\_am).

Next Vbid\_pr is replicated 8 times to create the matrix Mbid\_pr. The vector Vbid\_pr has the orders’d bid prices per customer.

The approach for this model is to find the fixed costs first. I follow this strategy because we have some restrictions and fixed cost in our problem. When one plant produces an order it has 70000 euros fixed cost each month. Also a plant can produce at most 120kg of the supplement per month at a cost of 10 euros per kilogram(kg). Also the costs depends from the distance between plant and city(customer). The delivery costs 0.01 euro per kilometer(km).

For that reason the creation of an equation for all fixed costs is very good idea (F\_costs). Create a matrix with all costs. Subtracte from customers’ bid price per order, the order amount per order multipled by the cost per kg(10) and the cost for delivery(order amount(kg) multipled by city distances(km) multipled by 0.01 euros per km. The mathematical equation for this matrix is:

Last task is to create the equations for the model.

Objective:

subject to:

R input using IpSolve:

#================================================

#Problem 3 - ompr & Ipsolve

#===============================================

#import the necesary packages for ompr and ipsolve

library(dplyr)

library(ROI)

library(ROI.plugin.glpk)

library(ompr)

library(ompr.roi)

library(lpSolve)

#===============================================

#Matrix with City Dinstances(kilometers-km)(Cdist)

C\_dist <- matrix(c(

0, 976, 1808, 1984, 3029, 1532, 206, 2657,

976, 0, 1198, 1043, 2105, 1383, 833, 1722,

1808, 1198, 0, 794, 1418, 1325, 1597, 1020,

1984, 1043, 794, 0, 1167, 2058, 1773, 829,

3029, 2105, 1418, 1167, 0, 2750, 2818, 391,

1532, 1383, 1325, 2058, 2750, 0, 1251, 2352,

206, 833, 1597, 1773, 2818, 1251, 0, 2435,

2657, 1722, 1020, 829, 391, 2352, 2435, 0), nrow=8, ncol=8, byrow = TRUE)

#================================================

#vector with order amount(kilograms-kg)(Vord\_am)

Vord\_am <- c(68, 41, 37, 54, 33, 39, 58, 51)

#matrix with order amount in kg per customer per plant (Mord\_am)

Mord\_am <- matrix(c(

68, 68, 68, 68, 68, 68, 68, 68,

41, 41, 41, 41, 41, 41, 41, 41,

37, 37, 37, 37, 37, 37, 37, 37,

54, 54, 54, 54, 54, 54, 54, 54,

33, 33, 33, 33, 33, 33, 33, 33,

39, 39, 39, 39, 39, 39, 39, 39,

58, 58, 58, 58, 58, 58, 58, 58,

51, 51, 51, 51, 51, 51, 51, 51),nrow=8,ncol=8, byrow = TRUE)

Mord\_am

#================================================

#vector with order bid price (Vbidpr)

Vbid\_pr <- c(75840, 44470, 46420, 87880, 43950, 21100, 74950, 84080)

#================================================

#Matrix with order bid price (Mbidpr)

Mbid\_pr <- matrix(c(

75840, 75840, 75840, 75840, 75840, 75840, 75840, 75840,

44470, 44470, 44470, 44470, 44470, 44470, 44470, 44470,

46420, 46420, 46420, 46420, 46420, 46420, 46420, 46420,

87880, 87880, 87880, 87880, 87880, 87880, 87880, 87880,

43950, 43950, 43950, 43950, 43950, 43950, 43950, 43950,

21100, 21100, 21100, 21100, 21100, 21100, 21100, 21100,

74950, 74950, 74950, 74950, 74950, 74950, 74950, 74950,

84080, 84080, 84080, 84080, 84080, 84080, 84080, 84080),nrow=8, ncol=8, byrow = TRUE)

Mbid\_pr

#================================================

#create a matrix with fixed costs

#customer's bid price(Mbid\_pr) - [customer's order amount(Mord\_am)multipled by 10 euros per kilo]

# - [customer's order amount(Mord\_am)multiple with cities distances] multiple with[0,01(cost per km)]

#with this matrix we have the costs of transformation and production per order: Fixed Costs(F\_costs)

F\_costs <- Mbid\_pr - 10\*Mord\_am - (0.01\*(Mord\_am\*C\_dist))

F\_costs

#================================================

#ipsolve model

#set the objective equation

#use the fixed costs and add the production cost per plant(-70000euros)

objective = c(as.vector(F\_costs),rep(-70000,8))

objective

#build the constraint matrix with left hands of the equations

const\_mat = matrix(c(68, 41, 37, 54, 33, 39, 58, 51,rep(0,56), -120,rep(0,7),rep(0,8),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,49), -120,rep(0,6),rep(0,16),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,42), -120,rep(0,5),rep(0,24),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,35), -120,rep(0,4),rep(0,32),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,28), -120,rep(0,3),rep(0,40),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,21), -120,rep(0,2),rep(0,48),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,14), -120,0,rep(0,56),

68, 41, 37, 54, 33, 39, 58, 51,rep(0,7), -120,

rep(c(1,0,0,0,0,0,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,1,0,0,0,0,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,1,0,0,0,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,0,1,0,0,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,0,0,1,0,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,0,0,0,1,0,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,0,0,0,0,1,0),8),0,0,0,0,0,0,0,0,

rep(c(0,0,0,0,0,0,0,1),8),0,0,0,0,0,0,0,0),nrow=16, byrow = TRUE)

const\_mat

#create the matrix for the equality/inequality signs

const\_dir = c(rep("<=",8),rep("<=",8))

const\_dir

#create a vector with right hands of the equations

const\_rhs = c(rep(0,8),rep(1,8))

const\_rhs

#solve the problem

lp("max",objective, const\_mat, const\_dir, const\_rhs, all.bin = TRUE)

sol=lp("max",objective, const\_mat, const\_dir, const\_rhs, all.bin = TRUE)$solution

#=======================================================

#ready code from the lectures

#it presents the solutions and the max profit

Narr = 2 #Number of arrays of decision variables in the model

v = c("x","y") #Variable name (letter) of each array

Rows = c(8,8) #Number of rows in each array

Cols = c(8,1) #Number of columns in each array

rangeU = c(rep(0,Narr)) # value of k where an array ends

for(n in 1:Narr){

for(m in 1:n){ # counting number of vars up to that upper range value

rangeU[n] = rangeU[n] + Rows[m]\*Cols[m]

}

}

for(k in 1:length(sol)){

a = Narr # array number not yet known; initialized to Narr

notFound = TRUE # array not yet found

n = Narr-1 # initializing n before the loop

if(sol[k]!=0){

while((notFound)&(n>=1)){

if(k>rangeU[n]){

a = n+1

notFound = FALSE

} else{

a = n

n = n-1

}

}

if(a==1){ # p is the location of k within the array it lies, when counting row by row

p = k

} else{

p = k-rangeU[a-1]

}

i = floor(p/Cols[a])+1

j = p - Cols[a] \* floor(p/Cols[a])

if(j==0){

i = i-1

j <- Cols[a]

}

if(Rows[a]==1){

print(paste("var indexed ",k," is ",v[a],",",j," = ",sol[k]),quote=FALSE)

} else if(Cols[a]==1){

print(paste("var indexed ",k," is ",v[a],",",i," = ",sol[k]),quote=FALSE)

} else {

print(paste("var indexed ",k," is ",v[a],",",i,",",

j," = ",sol[k]),quote=FALSE)

}

}

}

print("The solution values of the remaining variables are zero.")

R output using IpSolve:

Success: the objective function is 240898.4

[1] var indexed 1 is x , 1 , 1 = 1

[1] var indexed 8 is x , 1 , 8 = 1

[1] var indexed 18 is x , 3 , 2 = 1

[1] var indexed 19 is x , 3 , 3 = 1

[1] var indexed 21 is x , 3 , 5 = 1

[1] var indexed 52 is x , 7 , 4 = 1

[1] var indexed 55 is x , 7 , 7 = 1

[1] var indexed 65 is y , 1 = 1

[1] var indexed 67 is y , 3 = 1

[1] var indexed 71 is y , 7 = 1

[1] "The solution values of the remaining variables are zero."

Outcome: The company’s COO should keep open the plants 1,3,7 (A,C,G). Also these plants should produce specific orders to reach the maximum profit of 240,898.4 euros.

* Plant 1(A) will serve Customer(Order) 1(A)

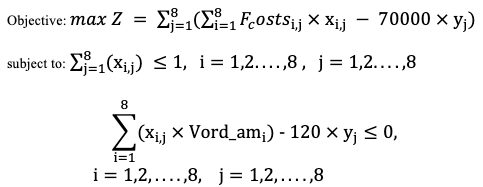
119kg

* Plant 1(A) will serve Customer(Order) 8(H)
* Plant 3(C) will serve Customer(Order) 2(B)
* Plant 3(C) will serve Customer(Order) 3( C)
* Plant 3(C) will serve Customer(Order) 5(E )

111kg

* Plant 7(G) will serve Customer(Order) 4(D)
* Plant 7(G) will serve Customer(Order) 7(G)

112kg

R input using ompr:

MIPModel() %>%

#================================================

#decision variables and set the type as binary

#i=orders, j=plants

add\_variable(x[i,j], i=1:8, j=1:8, type = "binary") %>%

add\_variable(y[j], j=1:8, type = "binary") %>%

#================================================

#objective function to find the max profit

#double sum expression

set\_objective(sum\_expr(sum\_expr(F\_costs[i,j]\*x[i,j], i=1:8)-70000\*y[j], j=1:8), "max") %>%

#================================================

#constraints

add\_constraint(sum\_expr(x[i,j],j=1:8)<=1, i=1:8) %>%

#constraints for order amount(kg)

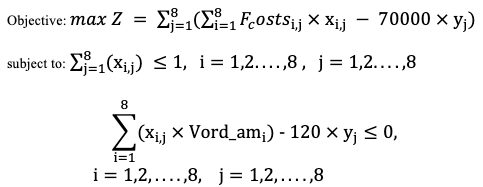
add\_constraint(sum\_expr(x[i,j]\*Vord\_am[i],i=1:8)-120\*y[j]<=0, j=1:8) %>%

#solve the model and find the max profit and plants and orders

solve\_model(with\_ROI(solver = "glpk", verbose=TRUE)) %>%

get\_solution(x[i,j]) %>%

filter(value>0)

R output using ompr:

<SOLVER MSG> ----

GLPK Simplex Optimizer, v4.65

16 rows, 72 columns, 136 non-zeros

\* 0: obj = -0.000000000e+00 inf = 0.000e+00 (64)

\* 39: obj = 2.546700000e+05 inf = 9.211e-16 (0)

OPTIMAL LP SOLUTION FOUND

GLPK Integer Optimizer, v4.65

16 rows, 72 columns, 136 non-zeros

72 integer variables, all of which are binary

Integer optimization begins...

Long-step dual simplex will be used

+ 39: mip = not found yet <= +inf (1; 0)

+ 71: >>>>> 1.668979500e+05 <= 2.540517100e+05 52.2% (18; 0)

+ 101: >>>>> 1.915674900e+05 <= 2.540109800e+05 32.6% (28; 1)

+ 340: >>>>> 1.970652400e+05 <= 2.534261500e+05 28.6% (117; 8)

+ 635: >>>>> 2.388615200e+05 <= 2.532320400e+05 6.0% (218; 27)

Solution found by heuristic: 240645.06

Solution found by heuristic: 240898.39

+ 10989: mip = 2.408983900e+05 <= tree is empty 0.0% (0; 4429)

INTEGER OPTIMAL SOLUTION FOUND

<!SOLVER MSG> ----

variable i j value

1 x 1 1 1

2 x 8 1 1

The maximum profit for the company will be 240,898.4 euros

3 x 2 3 1

4 x 3 3 1

5 x 5 3 1

6 x 4 7 1

7 x 7 7 1

Exactly the same results with ipsolve model.

Plant A – Customer A

Plant A – Customer H

Plant C – Customer B

Plant C – Customer C

Plant C – Customer E

Plant G – Customer D

Plant G – Customer G

# Problem 4

In this problem we have a term in the objective function (x1) which you present in our model with absolute value. The rest of the model is reminiscent of a normal system as easy to solve as those in problem 1 and 2.

The big difference here is the absolute value in term x1. What the absolute value does is that it allows the serum to take negative and positive values. This affects the solution of the model, which must be governed by some constraints in order for the whole model to be valid and solved as a linear program.

The solution I suggest is that in order for the system to be solved and its minimum value to be found, it must: