LINEAR PROGRAMMING MODELS

ALY6050: Introduction to Enterprise Analytics

DHRUV VIJAY GUJRATHI

Northeastern University

DHRUV VIJAY GUJRATHI, College of Professional Studies, Northeastern University, Boston,

MA 02115.

This report is regarding linear programming model project.

Dhruv Gujrathi is now a student at Department of Analytics, Northeastern University

Contact: gujrathi.d@northeastrn.edu

(NUID: 001029464)

INTRODUCTION

Linear programming, mathematical modeling technique in which a linear function is maximized or minimized when subjected to various constraints. This technique has been useful for guiding quantitative decisions in business planning, in industrial engineering, and—to a lesser extent—in the social and physical sciences. The solution of a linear programming problem reduces to finding the optimum value (largest or smallest, depending on the problem) of the linear expression (called the objective function)

$$f = c_1x_1 + \ldots + c_nx_n$$

The *a*'s, *b*'s, and *c*'s are constants determined by the capacities, needs, costs, profits, and other requirements and restrictions of the problem. The basic assumption in the application of this method is that the various relationships between demand and availability are linear; that is, none of the x_i is raised to a power other than 1. In order to obtain the solution to this problem, it is necessary to find the solution of the system of linear inequalities (that is, the set of n values of the variables x_i that simultaneously satisfies all the inequalities). The objective function is then evaluated by substituting the values of the x_i in the equation that defines f.

ANALYSIS

This report is regarding the profit and cost model analysis using linear programming formulation and analysis. It is focused on finding the maximised profit and inventory levels that can be stocked at a warehouse that is being rented by a company that we are working with. The company wants to find if the investment in the warehouse is going to get them good profits and how much stock or inventory should they store in the warehouse to get maximum profits. But, they have some constraints that we need to take care of. These factors include the area where the items will be stored, the dimensions of the shelves, the profit amount against every item and sales percentages required to determine the profits.

1. In the first question, we form the mathematical formulation of the problem. The mathematical formulation can be done using the items and the corresponding costs of the particular item. The first costs table that is given shows the cost price incurred by the company including transport to the warehouse and the second table shows the selling prices per unit to the customers which can be both, retail as well as distributors. The mathematical formulation depends on the conditions and constraints given by the company. These constraints include the selling price of each product or item. The purchasing monthly budget allotted by the company. The available stacking or storing space to store the items. The size of the pallets that are used to store the items. And lastly, the budget allotment given by the marketing department of the company.

   The mathematical formula to maximize the profits is:
   First of all, to calculate the profit against each item, we need to calculate it by subtracting the selling price from the cost price.
   Thus, the maximum profit, max z = 169.99P+359.99K+289.99G+134.99W.
   The constraints can be calculated as follows,
   - Cost Price Constraint: 330P+370K+410G+135W <=170000
   - Pallet dimensions Constraint: 5P+8K+5G+W <=2460.
   - Sales profit constraint: P>= 0.3(P+K+G+W)
     P>=0.3P+0.3K+0.3G+0.3W
     0.7P-0.3K-0.3G-0.3W >=0
   - Sales of generators and water pumps

2. The linear programming formulation in R is done by using the linear formulation package. The linear formulation is distributed into the form of a matrix, adding constraints as the variables in the matrix and then applying the linear formulation or programming to get the maximum profit, inventory value of each item and the sensitivity range of each sales profit value.

3. The linear programming formulation in R is done by using the linear formulation package called as LpSolveAPI or LPsolve. The report generated gives us the entire package contents pertaining to the vector on which linear programming has to be performed. The lpSolveAPI package provides an R interface to 'lp_solve', a Mixed Integer Linear Programming (MILP) solver with support for pure linear, (mixed) integer/binary, semi-continuous and special ordered sets (SOS) models.

4. An optimal solution to a linear program is the feasible solution with the largest objective function value (for a maximization problem). This means that the maximised profit and the maximum number of each item that can be stored in the warehouse is the optimal solution for our report. The optimal solution depends on the objective values which in this case were the profit values and secondly, the constraint values. The constraint values are

provided by the company and the profit values are calculated by subtracting the cost price from the selling price. The maximum profit calculated after applying all the constraints is $121,850.1 and the inventory level of all the four products is:

Sr. No.	Item	Inventory Level Value
1	Pressure Washer	171
2	Go-Kart	171
3	Generator	0
4	Case of 5 water pumps	228

From the above table, we can conclude that according to the dimension constraints and the shelf sizes given, 171 pressure washers, 171 Go-karts and 228 cases of 5 water pumps can be stored in the warehouse by the company.

Considering the above optimal solutions for the profit values and the no. of items to be stored, we can say that the company needs to allot some extra space to store the generators as the inventory level of that item has come out to be zero. Considering the profit value of the company after all the constraints are applied, they need not consider increasing the budget.

5. The optimal value of inventory level of stock item of ‘generator’ is zero. This is because of the constraints that haven applied to the objective function that contains the profit values of each items. The profit values are calculated by subtracting the cost price from the selling price. The sensitivity report of the objective function gives us the range of the profit values. This means that it gives us the upper limit value and the lower limit value of the item generator, so that the constraints applied to the objective function remain to zero. To change the optimal value of the inventory level of the item, the profit value has to be a value that is greater than the allocated profit value and the upper limit value in the sensitivity report. Thus, it has to be a value that is greater than the sum of 289.99 and 333.1. Thus, calculating the value at which the optimal solution of the inventory value of generators change to a non-zero value is 334$.
   Thus, concluding that if generators have to be stored in the warehouse and the constraint values have to be met, then the minimum inventory level of the generators has to be 20 nos. and the profit value will be $334.

6. To understand whether the company needs to allocate any more funds for the in addition to the already allocated $170,000, we need to understand the need of money or the expenditure that has been done in purchasing the number of goods to be stored in that warehouse. Now, we know the number of each item that can be stored in the warehouse according to the dimensions and area available. Thus, calculating the expenses of the item price (assuming the external expenses like electricity, rent, etc. is paid be the company as extra),

   Pressure Washers: 171 nos. x $330 = $56,430

   Go – Karts: 171 nos. x $370 = $63,270

   Generators: 0 nos. x $410 = $0

   Case of 5 Water Pumps: 229 nos. x 135 = 30,915

   Total Inventory = 571 nos.

   Total Expense: $150,615

   Difference Amount: $170,000 - $150615 = $19,385

   From the above calculations, we can conclude that we have not used the whole amount that is previously allocated and thus, there is no need for the company to allocate any more funds.

7. Looking at all the possibilities of renting a bigger or a smaller warehouse, we can say that the warehouse area can be a bit bigger as compared to the current area of the warehouse. This can be done by changing the per square feet area from 2460 to a value bigger than the upper limit of the sensitivity report. We also need to consider the fact that we need to be constant on the amount of profit that the company is earning. Thus, we add the upper sensitivity value to the total area and calculate the minimum value at which the profit value increases or decreases. Thus, the warehouse area can be increased from 2460 to 2768.89 to increase the inventory and the profit value.

CONCLUSION

We conducted a linear programming analysis formulation to determine the maximum profit and maximum inventory space that can be used in the warehouse. We can conclude on the basis of all the constraints put on by the company that they don’t need any more funds to run the warehousing facility. They might need some extra space to store some extra items to increase the profit. To come up with this, we have used the linear programming formulation model according to the constraints specified.

REFERENCES:

1. https://cran.r-project.org/web/packages/nloptr/nloptr.pdf
2. https://cran.r-project.org/web/packages/lpSolveAPI/lpSolveAPI.pdf
3. https://www.britannica.com/science/linear-programming-mathematics

R-Code:

```
#ALY6050_Project5_Gujrathi_D

#Question No. 2, 3, 4

#Intalling Package 'lpSolveAPI'

install.packages("lpSolveAPI")

#Calling library

library(lpSolveAPI)

#Setting the constraints and decision valriables

lprec <- make.lp(0,4)

lprec

#Getting the componets of variables

lp.control(lprec, sense="max")

#Setting the profits

set.objfn(lprec, c(169.99, 359.99, 289.99, 134.99))

#Setting cost price constraints

add.constraint(lprec, c(330, 370, 410, 135), "<=", 170000)

#Setting the dimension constraints

add.constraint(lprec, c(5, 8, 5, 1), "<=", 2460)

#Setting the profit percentage constraints for Pressure washers

add.constraint(lprec, c(0.7, -0.3, -0.3, -0.3), ">=", 0)

#Setting the profit percentage constraints for Generators

add.constraint(lprec, c(-0.3, 0.7, -0.3, -0.3), ">=", 0)

#Setting the profit percentage constraints for Go-Karts and Water pumps

add.constraint(lprec, c(0, 0, -2, 0.2), ">=", 0)

#Calling the vector with all constraints and decision variables

lprec

#Running summary of the vector
```

```
summary(lprec)

#Checking the vector

solve(lprec)

#Determining the Profit each month

get.objective(lprec)

#Determining the inventory value of each product

get.variables(lprec)

#Getting the sensitivity range for each product's profit value

get.sensitivity.obj(lprec)


#Question No. 5

#Changingthe profit value of the corresponding varibale whose optimum value is zero to obtain a non-sero value

set.objfn(lprec, c(169.99, 359.99, 334, 134.99))

add.constraint(lprec, c(330, 370, 410, 135), "<=", 170000)

add.constraint(lprec, c(5, 8, 5, 1), "<=", 2460)

add.constraint(lprec, c(0.7, -0.3, -0.3, -0.3), ">=", 0)

add.constraint(lprec, c(-0.3, 0.7, -0.3, -0.3), ">=", 0)

add.constraint(lprec, c(0, 0, -2, 0.2), ">=", 0)

lprec

summary(lprec)

solve(lprec)

get.objective(lprec) #proft

get.variables(lprec) #No. of each product

get.sensitivity.obj(lprec)
```

OUTPUT:

```
> #ALY6050_Project5_Gujrathi_D
> #Question No. 2, 3, 4
> #Intalling Package 'lpSolveAPI'
> install.packages("lpSolveAPI")
Error in install.packages : Updating loaded packages

Restarting R session...

> install.packages("lpSolveAPI")
WARNING: Rtools is required to build R packages but is not currently installed. Please download an
d install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘E:/data 3-5-19/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/lpSolveAPI_5.5.2.0-17.7.zip'
Content type 'application/zip' length 965048 bytes (942 KB)
downloaded 942 KB

package ‘lpSolveAPI’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\admin\AppData\Local\Temp\Rtmp8watWo\downloaded_packages
> #Calling library
> library(lpSolveAPI)
> #Setting the constraints and decision valriables
> lprec <- make.lp(0,4)
> lprec
Model name:
            C1    C2    C3    C4
Minimize     0     0     0     0
Kind       Std   Std   Std   Std
Type      Real  Real  Real  Real
Upper      Inf   Inf   Inf   Inf
Lower        0     0     0     0
> #Getting the componets of variables
> lp.control(lprec, sense="max")
$anti.degen
[1] "none"

$basis.crash
[1] "none"

$bb.depthlimit
[1] -50

$bb.floorfirst
[1] "automatic"

$bb.rule
[1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"

$break.at.first
```

```
[1] FALSE

$break.at.value
[1] 1e+30

$epsilon
      epsb       epsd      epsel     epsint epsperturb   epspivot
     1e-10      1e-09      1e-12      1e-07      1e-05      2e-07

$improve
[1] "dualfeas" "thetagap"

$infinite
[1] 1e+30

$maxpivot
[1] 250

$mip.gap
absolute relative
   1e-11    1e-11

$negrange
[1] -1e+06

$obj.in.basis
[1] TRUE

$pivoting
[1] "devex"    "adaptive"

$presolve
[1] "none"

$scalelimit
[1] 5

$scaling
[1] "geometric"   "equilibrate" "integers"

$sense
[1] "maximize"

$simplextype
[1] "dual"   "primal"

$timeout
[1] 0

$verbose
[1] "neutral"

> #Setting the profits
> set.objfn(lprec, c(169.99, 359.99, 289.99, 134.99))
> #Setting cost price constraints
```

```
> add.constraint(lprec, c(330, 370, 410, 135), "<=", 170000)
> #Setting the dimension constraints
> add.constraint(lprec, c(5, 8, 5, 1), "<=", 2460)
> #Setting the profit percentage constraints for Pressure washers
> add.constraint(lprec, c(0.7, -0.3, -0.3, -0.3), ">=", 0)
> #Setting the profit percentage constraints for Generators
> add.constraint(lprec, c(-0.3, 0.7, -0.3, -0.3), ">=", 0)
> #Setting the profit percentage constraints for Go-Karts and Water pumps
> add.constraint(lprec, c(0, 0, -2, 0.2), ">=", 0)
> #Calling the vector with all constraints and decision variables
> lprec
Model name: 
            C1      C2      C3      C4         
Maximize  169.99  359.99  289.99  134.99         
R1         330     370     410     135  <=  170000
R2           5       8       5       1  <=    2460
R3         0.7    -0.3    -0.3    -0.3  >=       0
R4        -0.3     0.7    -0.3    -0.3  >=       0
R5           0       0      -2     0.2  >=       0
Kind       Std     Std     Std     Std         
Type      Real    Real    Real    Real         
Upper      Inf     Inf     Inf     Inf         
Lower        0       0       0       0         
> #Running summary of the vector
> summary(lprec)
    Length      Class       Mode 
         1   lpExtPtr externalptr 
> #Checking the vector
> solve(lprec)
[1] 0
> #Determining the Profit each month
> get.objective(lprec)
[1] 121850.1
> #Determining the inventory value of each product
> get.variables(lprec)
[1] 171.6279 171.6279   0.0000 228.8372
> #Getting the sensitivity range for each product's profit value
> get.sensitivity.obj(lprec)
$objfrom
[1]  1.544000e+01  2.054400e+02 -1.000000e+30  1.035561e+02

$objtill
[1] 3.962674e+02 5.979127e+02 3.331202e+02 1.000000e+30

> #Question No. 5
> #Changingthe profit value of the corresponding varibale whose optimum value is zero to obtain a no
n-sero value
> set.objfn(lprec, c(169.99, 359.99, 334, 134.99))
> add.constraint(lprec, c(330, 370, 410, 135), "<=", 170000)
> add.constraint(lprec, c(5, 8, 5, 1), "<=", 2460)
> add.constraint(lprec, c(0.7, -0.3, -0.3, -0.3), ">=", 0)
> add.constraint(lprec, c(-0.3, 0.7, -0.3, -0.3), ">=", 0)
> add.constraint(lprec, c(0, 0, -2, 0.2), ">=", 0)
> lprec
Model name:
```

```
           C1     C2     C3     C4
Maximize  169.99  359.99    334  134.99
R1          330    370    410    135  <=  170000
R2            5      8      5      1  <=    2460
R3          0.7   -0.3   -0.3   -0.3  >=       0
R4         -0.3    0.7   -0.3   -0.3  >=       0
R5            0      0     -2    0.2  >=       0
R6          330    370    410    135  <=  170000
R7            5      8      5      1  <=    2460
R8          0.7   -0.3   -0.3   -0.3  >=       0
R9         -0.3    0.7   -0.3   -0.3  >=       0
R10           0      0     -2    0.2  >=       0
Kind        Std    Std    Std    Std
Type       Real   Real   Real   Real
Upper       Inf    Inf    Inf    Inf
Lower         0      0      0      0
> summary(lprec)
    Length      Class        Mode
         1   lpExtPtr externalptr
> solve(lprec)
[1] 0
> get.objective(lprec) #proft
[1] 121867.8
> get.variables(lprec) #No. of each product
[1] 166.0123 166.0123  20.1227 201.2270
> get.sensitivity.obj(lprec)
$objfrom
[1] -102.09863 -305.19825  333.12023   51.06407

$objtill
[1] 1.731425e+02 3.631425e+02 1.000000e+30 1.356312e+02
```