Linear Programming in Restaurant Business

Cluster Innovation Centre

University of Delhi - 110007

Submitted By:

Hrithik Kumar Verma

Mohammad Abdullah

Saumya Kumari

April 2021

Month-Long Project Report submitted for the paper

VI.1 Linear Construction of Actions: Engineering through Linear Programming and Game Theory

in the partial fulfilment for the degree of

B.Tech (Information Technology and Mathematical Innovations)

Acknowledgement

We would like to express our sincere gratitude towards Dr. Nirmal Yadav for providing us with an opportunity to work on this project 'Linear Programming in Restaurant Business'. Her learned advice and constant encouragement have helped us complete this project.

We would also like to thank our team members for their collaborative work, suggestions, contributions and guidance throughout the project.

Abstract

Linear Programming in Restaurant Business.

by

Hrithik Kumar Verma

Mohammad Abdullah

Saumya Kumari

Cluster Innovation Centre, 2021

Linear programming, an operation research technique is widely used in finding solutions to complex managerial decision problems. There are problems for maximizing the profits, problems for minimizing the cost, etc. Several advancements in industries involving the optimal use of resources and efficient processes has only been the result of Linear Programming. Every domain including food and industry, agriculture, engineering, transportation, manufacturing, and energy implements the methodologies of operations research to make their processes and profits efficient and high, respectively. Restaurant business is being groomed day by day where these technologies play a pivotal role to make the restaurant management smooth and its operations better. In our project, we have taken our own hypothetical restaurant which has just started its operations. Assuming that we don't have enough resources and manpower and along with that, a limited number of items, we then plan to gain a maximum profit from our daily investment. Having said that, we try to explore the plethora of Linear Programming and have used tools such as Python Programming using Pulp, and Microsoft Excel LP Solver add-ins to reach our objective function of maximizing the sales profit and analyse the various user case scenarios.

Table of Contents

1. Introduction

Linear programming is a management method used by most organizations, large and small, that have a finite supply of resources. It is concerned with the optimization of a well-defined objective function, which is subject to a collection of linear equations and/or inequalities known as constraints.

Any organization, corporation, or firm strives to make a profit because it is the only way to ensure its continued survival and productivity. Manufacturing companies at all levels face the challenge of producing products (cars, machines, clothing, breads, etc.) of the right quality, quantity, and time, and, more importantly, at the lowest cost (minimized cost) and highest benefit for their sustainability and development in today's world. As a result, the industry's productive productivity must improve. The aim of this project is to use linear programming (LP) as a mathematical model to maximize benefit in manufacturing industries like restaurant business.

1.1 Operations Research

Operations Research is a scientific decision-making method that aims to find the best design and operation of a system, normally under circumstances where finite resources must be allocated. To make the best choice, the scientific method involves the use of one or more mathematical/optimization models (i.e., representations of the actual situation).

An optimization model tries to find the values of the decision variables that best optimize (maximize or minimize) an objective function from a collection of all possible values for the decision variables that satisfy the constraints. The following are the three major components:

- A function that needs to be refined is called an objective function (maximized or minimized)

- Decision variables are variables that can be regulated and affect the system's output.

- Constraints are a collection of constraints on decision variables (such as linear inequalities or equalities). The decision variables are limited to taking positive values by a non-negativity constraint (e.g. you cannot produce a negative number of items x1, x2 and x3).

The solution of the optimization model is called the optimal feasible solution.

1.2 Linear Programming

When all of the goals and constraints are linear (in the variables) and all of the decision variables are continuous,linear programming generates an optimal solution. Linear programming (also known as LP) is an operations analysis technique. Linear programming is the simplest operations analysis method in terms of hierarchy.

Linear programming is the process of using mathematical models to solve linear problems in order to maximize or minimize an objective function while keeping some constraints in mind. As a result, it can be applied to any problem that can be interpreted as a linear function with parameters and constraints. The well-known knapsack and traveling salesman problems, for example, are optimization problems that can be solved with linear programming. The algorithm's main goal will be to find the best values for all of the parameters of a function Z (Objective Function) in order to find the best fit for our problem.

Hopefully, we already have algorithms (e.g., the simplex algorithm) to solve this, and as developers, we can also use solvers that are specifically optimized to solve these types of problems. Most programming languages come with solver libraries that are simple to use. The difficult part is figuring out how to convert our initial problem into a function with parameters and constraints, rather than using the algorithm itself that will solve our problem.

1.3 Simplex Method

The Simplex method is an approach to solving linear programming models by hand using slack variables, tableaus, and pivot variables as a means to finding the optimal solution of an optimization problem. A linear program is a method of achieving the best outcome given a maximum or minimum equation with linear constraints. Most linear programs can be solved using an online solver such as MatLab, but the Simplex method is a technique for solving linear problems by hand. To solve a linear programming model using the Simplex method the following steps are necessary:

- Standard form
- Introducing slack variables
- Creating the tableau
- Pivot variables
- Creating a new tableau
- Checking for optimality
- Identify optimal value

2. Problem Description

Eating is an important part of our daily lives. Busy people often struggle to find time for this task, so they either expect to eat at restaurants where they won't have to wait long or to be taught how to cook efficiently at home. In any case, developing procedure guidelines that adheres to all operating rules and equipment availability in a kitchen setting while still completing dishes as quickly as possible is critical. In most restaurants, this job is left to the discretion of professional chefs.This worked in the past, but with more orders coming in every day and the need to reduce operating costs, some restaurants are turning to operational research to find a solution. We hope to focus on the benefit maximization issue in order to create an automated program for constructing productive procedure guidances in light of these demands and the progress made in the restaurant industry. The hypothetical restaurant that we created implements the same set of guidelines and procedures and generates a good sale through an automated program set up.

In this project, we have a local hypothetical restaurant which has been newly built, inaugurated and opened. Our team is unaware of the tastes of the local people, and hence, for the time being we have limited our small restaurant to only serving not more than 10 items. We have decided the items based on the preference of local people by observing and analysing the needs of the people and at the same time, keeping in mind the present trends. Mostly, the restaurant deals in serving fast food or snacks to the customers.

3. Formulating the Problem

We need to formulate our business plan mathematically, and then solve it through Linear Programming. And the way that we use Linear Programming can be in two ways:

1. <u>Writing a Python Program(Using Pulp Library)</u>

2. <u>Solving with Microsoft Excel Solver.</u>

As already said, the restaurant is producing 10 food items. The main objective is to maximize the profits earned from the sale of these food items. The food items that has been decided are as follows:

1. Burger 2. French Fries 3. Pizza 4. Cold Drinks

5. Pasta 6. Salad 7. Momos 8. Ice-cream

9. Sandwich 10. Samosa

These are the food items that are meant to be served at the restaurant. The production of each of the items requires a certain amount of cost and time. We need a certain amount of money, labour and time to produce each of these 10 items listed above. Costs such as rent of the restaurant, raw materials needed to prepare these items such as buns, pizza base, vegetables, corn flour, artificial additives and much more, machines such as an oven, and steamer, etc. Combining all these costs, we have calculated an approximated estimate of what the production price would be for each of the items listed above. And to sustain the restaurant for a long time, and to pay the chefs or staff at the restaurant, a selling price has been decided for each of the products. And the profits calculated from the selling price and the production price has to be maximized. Each single item has been associated with a decision variable below and our main objective is to find the exact number of every single item that needs to be produced so that the profit can be maximized.

The available resources are:

- 8 Chefs/staff working for 12 hours per day.
- The maximum investment per day is Rs. 20,000/-
- Maximum number of Appliances and Utensils it has is 350.

3.1 Decision Variables

The decision variables are variables that reflect (unknown) decisions that need to be taken. Problem data, on the other hand, are values that are either given or can be easily calculated from what is given. For this problem, the decision variables are the food items on the menu of a restaurant that needs to be produced in order to maximize profit.
These unknown values will be denoted by x1, x2, x3, x4, x5, x6, x7, x8, x9 and x10, respectively.

Following are the food items on the menu of a restaurant:

x1	Burger
x2	French Fries
x3	Pizza
x4	Cold Drinks
x5	Pasta
x6	Salad
x7	Momos
x8	Ice-cream
x9	Sandwich
x10	Samosa

3.2 Production Cost

Sl. No	**Items**	**Cost Price**	**Selling Price**	**Profit**
1.	Burger (x1)	150	220	*70*
2.	French Fries (x2)	90	140	*80*

3.	Pizza (x3)	250	350	*100*
4.	Cold Drink (x4)	95	140	*45*
5.	Pasta (x5)	110	150	*40*
6.	Salad (x6)	50	95	*45*
7.	Momos (x7)	75	130	*55*
8.	Ice Cream (x8)	80	100	*20*
9.	Sandwich (x9)	40	90	*50*
10	Samosa (x10)	20	35	*15*

3.3 Objective Function

All linear programming problems aim to either maximize or minimize some numerical value representing profit, cost, production quantity, etc. It evaluates the amount by which each decision variable would contribute to the net present value of a project or an activity. In this case, our objective is to maximize the profits generated from the production cost.

We need to create an objective function Z which needs to be maximum at the end of this project. And the Objective function is as follows:

Maximize Z = 70x1 +80x2 + 100x3 + 45x4 + 40x5 + 45x6 + 55x7 + 20x8 + 50x9 + 15x10

The objective function can be easily maximized. Anyone would say, sell the item that gives the maximum profit and sell it unlimited. But no, this is not the solution. Because in a real world scenario, things are different. We have limited resources that are available to us. We have limited manpower available to us. Some things that need to be kept in mind will be discussed after this and later, these things will act as constraints on our model.

3.4 Cooking Time

The cooking time per item of the 10 items was listed on an observational basis. These were recorded as:

Items	Cooking time per unit (in minutes)
Burger (x1)	25
French Fries (x2)	20
Pizza (x3)	30
Cold Drink (x4)	5
Pasta (x5)	40
Salad (x6)	10
Momos (x7)	20
Ice Cream (x8)	7
Sandwich (x9)	10
Samosa (x10)	50

3.5 Appliances/Utensils Allotment

Our small restaurant has a specific set of appliances that can be used in the preparation of the above 10 items. Each item shares its own appliances or utensils to be prepared. Below is a record of the number of appliances that is utilized in per item preparation per unit.

Items	Appliances/Utensils needed per unit
Burger (x1)	1
French Fries (x2)	2
Pizza (x3)	2
Cold Drink (x4)	1
Pasta (x5)	3
Salad (x6)	4
Momos (x7)	3
Ice Cream (x8)	2
Sandwich (x9)	1
Samosa (x10)	2

3.6 Minimum Selling Conditions

It may happen in an optimization problem, that to increase the profit it may sell only a particular item so many times rather than having sold a mix of all items. But in reality, this does not happen. Even if an item is giving us a maximum profit but to adhere to a mixed customer need and a variety market, we need to ensure that the model that we are creating sells at least a predetermined minimum number of the items.

Items	Minimum Numbers to be sold
Burger (x1)	35
French Fries (x2)	16
Pizza (x3)	20
Cold Drink (x4)	11
Pasta (x5)	12
Salad (x6)	5
Momos (x7)	15
Ice Cream (x8)	30
Sandwich (x9)	18
Samosa (x10)	25

3.7 Other Conditions

There are certainly other limitations linked to the restaurant business. And these limitations are beneficial for the sustainability of the restaurant as well as for the market growth but at the same time, the production gets compromised. But these limitations eventually become necessary as per marketing strategies or for profit goals.

1. To ensure that a certain level is maintained while counting the profit amount, it is necessary to plan that all the items sold within a day must be at least 150. This is a necessary criteria to make the restaurant make itself in the long run and to pay for its employees.

2. The restaurant wants to make its name in the food market with Burger and Pizza, and hence it decides that whatever the total sale would be at the end of the day, at least 30 % of the items being sold should be Burger and Pizza.

3. The restaurant assumes that for a mix of burger and pizza, 3 cold drinks would be sold, surely! We all know people love cold drinks, with burgers and pizza.

3.8 Constraints

The restrictions or limitations on the total amount of a particular resource required to carry out the activities that would decide the level of achievement in the decision variables. These restrictions are known as constraints. In the standard form of a linear programming problem, all constraints are in the form of equations.
With respect to the given problem and the above conditions, we will have the following constraints:

- A maximum daily investment of Rs. 20,000/-. The costs of production per item per unit given for all items should be less than or equal to 20000.

Constraint 1

$$150x1 + 90x2 + 250x3 + 95x4 + 110x5 + 50x6 + 75x7 + 80x8 + 40x9 + 20x10 <= 20000$$

- The 8 workers working 12 hours a day should give us the maximum time in which the productivity of the items shall be continued and the restaurant shall be operational.
  This is on a per day basis.
  Hence, 8 * 12 = 96 Hours

  96 Hours = 96 * 60 = 5760 minutes

  Constraint 2

$$25x1 + 20x2 + 30x3 + 5x4 + 40x5 + 10x6 + 25x7 + 7x8 + 10x9 + 50x10 <= 5760$$

- The appliances and utensils are limited. Hence all appliances used per item per unit when totalled should be less than or equal to 350.

  Constraint 3

$$x1 + 2x2 + 2x3 + x4 + 3x5 + 4x6 + 3x7 + 2x8 + x9 + 2x10 <= 350$$

- A minimum sale of all items that is 150.

  Constraint 4

$$x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 >= 150$$

- For a mix of a burger and pizza, people will buy 3 drinks

Constraint 5

$$3x1 + 3x3 - 2x4 >= 0$$

- From all the sales, at least 30 % of the sales should be generated from burger and pizza since they are the market leaders in the food business.

Constraint 6

$$x1 + x3 >= 0.3x1 + 0.3x3 + 0.3(x2 + x4 + x5 + - - - + x10)$$

- A minimum selling criteria for all the items that is listed above in the table.

Constraint 7 - Constraint 16

$x1 >= 35$	$x6 >= 5$
$x2 >= 16$	$x7 >= 15$
$x3 >= 20$	$x8 >= 30$
$x4 >= 11$	$x9 >= 18$
$x5 >= 12$	$x10 >= 25$

3.9 Non – negativity restrictions

Regardless of whether the goal is to maximize or decrease the net present value of an operation, each decision variable in a Linear Programming model must be positive. This is a critical restriction.

Apart from making it just positive, we have already specified certain values to the decision variables as per requirement.

4. Final Model

This gives us the complete model of this problem:

Maximize Z

- 70x1 +80x2 + 100x3 + 45x4 + 40x5 + 45x6 + 55x7 + 20x8 + 50x9 + 15x10

Subject to the following Constraints :

- $150x1 + 90x2 + 250x3 + 95x4 + 110x5 + 50x6 + 75x7 + 80x8 + 40x9 + 20x10 <= 2000$
- $25x1 + 20x2 + 30x3 + 5x4 + 40x5 + 10x6 + 25x7 + 7x8 + 10x9 + 50x10 <= 5760$
- $x1 + 2x2 + 2x3 + x4 + 3x5 + 4x6 + 3x7 + 2x8 + x9 + 2x10 <= 350$
- $x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 >= 150$
- $3x1 + 3x3 - 2x4 >= 0$
- $x1 + x3 >= 0.3x1 + 0.3x3 + 0.3\,(x2 + x4 + x5 + \; - \; - \; - \; + x10)$
- $x1 >= 35$

- $x2 >= 16$

- $x3 >= 20$

- $x4 >= 11$

- $x5 >= 12$

- $x6 >= 5$

- $x7 >= 15$

- $x8 >= 30$

- $x9 >= 18$

- $x10 >= 25$

In Linear Programming, there are numerous ways to solve a Linear Programming Problem. We have tried to solve it with Excel Solver which is an add-ins in Microsoft Excel and is a very interactive tool which makes the work easier.

The other method implemented is doing it with Python using a library called Pulp. Pulp is a very powerful library meant for solving linear programming problems with Python. It has several functions which help us in Maximization or minimization, we can declare decision variables and constraints with the help of these functions to create a LPP model and then find if it's feasible or not. If feasible, then solve it.

5. Solution with Excel Solver

This is a setup in MS Excel declaring decision variables and constraints where the initial values of all decision variables that we have taken is 1. The “value” row has been sum multiplied in excel with the “sum multiply” function of excel with the “coefficients” row and the “constraints” row generating the “LHS” column.
This “LHS” column along with the “Objective Value” is then referenced as cells to the solver add-in from the data subsection.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1						Initial Random Values										
2																
3		Decision Variable	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Objective Value			
4		Value	1	1	1	1	1	1	1	1	1	1	520			
5		Coefficients	70	80	100	45	40	45	55	20	50	15				
6																
7																
8		Constraints											LHS		RHS	
9		Constraint 1	150	90	250	95	110	50	75	80	40	20	960	<=	20000	
10		Constraint 2	25	20	30	5	40	10	25	7	10	50	222	<=	5760	
11		Constraint 3	1	2	2	1	3	4	3	2	1	2	21	<=	350	
12		Constraint 4	1	1	1	1	1	1	1	1	1	1	10	>=	150	
13		Constraint 5	3		3	-2							4	>=	0	
14		Constraint 6	0.7	-0.3	0.7	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-1	>=	0	
15		Constraint 7	1										1	>=	35	
16		Constraint 8		1									1	>=	16	
17		Constraint 9			1								1	>=	20	
18		Constraint 10				1							1	>=	11	
19		Constraint 11					1						1	>=	12	
20		Constraint 12						1					1	>=	5	
21		Constraint 13							1				1	>=	15	
22		Constraint 14								1			1	>=	30	
23		Constraint 15									1		1	>=	18	
24		Constraint 16										1	1	>=	25	
25																
26																
27																

Fig 1 : Excel Screenshot of initial problem formulation

5.1 Deduced Outputs:

This yellow strip running through the values is the optimal solution showing the quantity of the different food items that need to be produced in order to gain a maximum profit and that maximum profit is given by the value of the “Objective Value”.

Burger = 38.0

French Fries = 16.0

Pizza = 20.0

Cold Drinks = 11.0

Pasta = 12.0

Salad = 5.0

Momos = 15.0

Ice Cream = 30.0

Sandwich = 18.0

Samosa = 25.0

Value of Maximize Objective Function is : Rs. 9840

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3		Decision Variabl	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Objective Value			
4		Value	38	16	20	11	12	5	15	30	18	25	9840			
5		Coefficients	70	80	100	45	40	45	55	20	50	15				
6																
7																
8		Constraints											LHS		RHS	
9		Constraint 1	150	90	250	95	110	50	75	80	40	20	19500	<=	20000	
10		Constraint 2	25	20	30	5	40	10	25	7	10	50	4470	<=	5760	
11		Constraint 3	1	2	2	1	3	4	3	2	1	2	350	<=	350	
12		Constraint 4	1	1	1	1	1	1	1	1	1	1	190	>=	150	
13		Constraint 5	3		3	-2							152	>=	0	
14		Constraint 6	0.7	-0.3	0.7	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	1	>=	0	
15		Constraint 7	1										38	>=	35	
16		Constraint 8		1									16	>=	16	
17		Constraint 9			1								20	>=	20	
18		Constraint 10				1							11	>=	11	
19		Constraint 11					1						12	>=	12	
20		Constraint 12						1					5	>=	5	
21		Constraint 13							1				15	>=	15	
22		Constraint 14								1			30	>=	30	
23		Constraint 15									1		18	>=	18	
24		Constraint 16										1	25	>=	25	
25																
26																
27																
28																
29																

Fig 2 : Excel Screenshot of final problem formulation

5.2 Solver Outputs:

1. <u>Answer Report:</u>

Microsoft Excel 16.0 Answer Report
Worksheet: [Book1]Sheet1
Report Created: 5/2/2021 8:36:50 PM
Result: Solver found a solution. All Constraints and optimality conditions are satisfied.
Solver Engine
Engine: Simplex LP
Solution Time: 0.063 Seconds.
Iterations: 20 Subproblems: 0
Solver Options
Max Time Unlimited, Iterations Unlimited, Precision 0.000001
Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
M4	Value Objective Value	520	9840

Variable Cells

Cell	Name	Original Value	Final Value	Integer
C4	Value X1	1	38	Contin
D4	Value X2	1	16	Contin
E4	Value X3	1	20	Contin
F4	Value X4	1	11	Contin
G4	Value X5	1	12	Contin
H4	Value X6	1	5	Contin
I4	Value X7	1	15	Contin
J4	Value X8	1	30	Contin
K4	Value X9	1	18	Contin
L4	Value X10	1	25	Contin

Fig 3 : Excel Answer Report

2. <u>Sensitivity Analysis Report:</u>

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
C4	Value X1	38	0	70	1E+30	20
D4	Value X2	16	0	80	60	1E+30
E4	Value X3	20	0	100	40	1E+30
F4	Value X4	11	0	45	25	1E+30
G4	Value X5	12	0	40	170	1E+30
H4	Value X6	5	0	45	235	1E+30
I4	Value X7	15	0	55	155	1E+30
J4	Value X8	30	0	20	120	1E+30
K4	Value X9	18	0	50	20	1E+30
L4	Value X10	25	0	15	125	1E+30

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
M11	Constraint 4 LHS	190	0	150	40	1E+30
M12	Constraint 5 LHS	152	0	0	152	1E+30
M13	Constraint 6 LHS	1	0	0	1	1E+30
M14	Constraint 7 LHS	38	0	35	3	1E+30
M15	Constraint 8 LHS	16	-60	16	0.588235294	2.380952381
M16	Constraint 9 LHS	20	-40	20	1.428571429	10
M17	Constraint 10 LHS	11	-25	11	1	9.090909091
M18	Constraint 11 LHS	12	-170	12	0.416666667	1.470588235
M19	Constraint 12 LHS	5	-235	5	0.322580645	0.909090909
M20	Constraint 13 LHS	15	-155	15	0.416666667	1.333333333
M21	Constraint 14 LHS	30	-120	30	0.588235294	2.272727273
M22	Constraint 15 LHS	18	-20	18	1	4.545454545
M23	Constraint 16 LHS	25	-125	25	0.588235294	1.785714286
M8	Constraint 1 LHS	19500	0	20000	1E+30	500
M9	Constraint 2 LHS	4470	0	5760	1E+30	1290
M10	Constraint 3 LHS	350	70	350	3.333333333	1.428571429

Fig 5 : Excel Screenshot of sensitivity report

6. Solution with PuLP library of python

PuLP

PuLP is a Python-based free open source project. It's a mathematical model for describing optimization problems. PuLP will then use python commands to control and show the solution, which can be solved using any of a number of external LP solvers (CBC, GLPK, CPLEX, Gurobi, and so on).

We will be writing programs in the Python Jupyter notebook importing the Pulp library and using its several inbuilt functions.

Import required Libraries

In [2]:
```
# import the library pulp as p
import pulp as p
```

Model Initialization

Define the problem by naming it appropriately; in this case, we called it "Maximization_of_Restaurant_Profit." Also, we added whether we want to Maximize or Minimize the objective function.

In [3]:
```
# Create a LP Minimization problem
prob = p.LpProblem("Maximization_of_Restaurant_Profit", p.LpMaximize)
```

Problem Description

We have already defined constraints extracted and formulated mathematically from the problem statement. All we need to do is to tell the Python interpreter that it is a decision variable and also specify what type of value it is and what is its lower bound or the minimum value.

```
In [7]: # Display the problem
        print(prob)

Maximization_of_Restaurant_Profit:
MAXIMIZE
70*x1 + 15*x10 + 80*x2 + 100*x3 + 45*x4 + 40*x5 + 45*x6 + 55*x7 + 20*x8 + 50*x9 + 0
SUBJECT TO
_C1: 150 x1 + 20 x10 + 90 x2 + 250 x3 + 95 x4 + 110 x5 + 50 x6 + 75 x7 + 80 x8
 + 40 x9 <= 20000

_C2: 25 x1 + 50 x10 + 20 x2 + 30 x3 + 5 x4 + 40 x5 + 10 x6 + 25 x7 + 7 x8
 + 10 x9 <= 5760

_C3: x1 + 2 x10 + 2 x2 + 2 x3 + x4 + 3 x5 + 4 x6 + 3 x7 + 2 x8 + x9 <= 350

_C4: x1 + x10 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 >= 150

_C5: 3 x1 + 3 x3 - 2 x4 >= 0

_C6: 0.7 x1 - 0.3 x10 - 0.3 x2 + 0.7 x3 - 0.3 x4 - 0.3 x5 - 0.3 x6 - 0.3 x7
 - 0.3 x8 - 0.3 x9 >= 0

VARIABLES
35 <= x1 Integer
25 <= x10 Integer
16 <= x2 Integer
20 <= x3 Integer
11 <= x4 Integer
12 <= x5 Integer
5 <= x6 Integer
15 <= x7 Integer
30 <= x8 Integer
18 <= x9 Integer
```

Defining Decision Variables

```
In [4]: # Create problem Variables

        items = {"x1": "Burger", "x2": "French Fries", "x3": "Pizza", "4": "Cold Drinks", "x5": "Pasta", "x6": "Salad", "x7": "Momos",
                 "x8": "Ice Cream", "x9": "Sandwich", "x10": "Samosa"}
        x1 = p.LpVariable("x1", lowBound=35, cat = 'Integer')
        x2 = p.LpVariable("x2", lowBound=16, cat = 'Integer')
        x3 = p.LpVariable("x3", lowBound=20, cat = 'Integer')
        x4 = p.LpVariable("x4", lowBound=11, cat = 'Integer')
        x5 = p.LpVariable("x5", lowBound=12, cat = 'Integer')
        x6 = p.LpVariable("x6", lowBound=5, cat = 'Integer')
        x7 = p.LpVariable("x7", lowBound=15, cat = 'Integer')
        x8 = p.LpVariable("x8", lowBound=30, cat = 'Integer')
        x9 = p.LpVariable("x9", lowBound=18, cat = 'Integer')
        x10 = p.LpVariable("x10", lowBound=25, cat = 'Integer')
```

Objective Function

```
In [5]: #Objective Function
prob += 70 * x1 + 80 * x2 + 100 * x3 + 45 * x4 + 40 * x5 + 45 * x6 + 55 * x7 + 20 * x8 + 50 * x9 + 15 * x10
```

Constraints

```
In [6]: #Constraints

# Max cost per item per day
prob += 150 * x1 + 90 * x2 + 250 * x3 + 95 * x4 + 110 * x5 + 50 * x6 + 75 * x7 + 80 * x8 + 40 * x9 + 20 * x10 <= 20000
# 8 workers work 12 hours per day
prob += 25 * x1 + 20 * x2 + 30 * x3 + 5 * x4 + 40 * x5 + 10 * x6 + 25 * x7 + 7 * x8 + 10 * x9 + 50 * x10 <= 5760
# Appliances and utensils used for all items per day is 350
prob += 1 * x1 + 2 * x2 + 2 * x3 + 1 * x4 + 3 * x5 + 4 * x6 + 3 * x7 + 2 * x8 + 1 * x9 + 2 * x10 <= 350
# Minimum no of sale should be 150
prob += x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 >= 150
# For a mix of a burger and pizza, people will buy 3 drinks
prob += 3 * x1 + 3 * x3 - 2 * x4 >= 0
# From total sales, 30 % sales should be at least of pizza and burger
prob += x1 + x3 >= 0.3 * (x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10)
```

6.1 Python Output:

```
Optimal
The Restaurant needs to produce the items in following quantities:

Burger   =    38.0
French Fries   =    16.0
Pizza   =    20.0
Cold Drinks   =    11.0
Pasta   =    12.0
Salad   =    5.0
Momos   =    15.0
Ice Cream   =    30.0
Sandwich   =    18.0
Samosa   =    25.0

And the maximum profit is : Rs.  9840.0

Process finished with exit code 0
```

Fig 6 : Python terminal output screenshot.

Conclusion

From both the methods of maximizing the profits from the sale of the restaurant, that is, from the Microsoft Excel Solver and from the Python Pulp Library, we have got the exact same results. This cements our belief that we have reached a correct operation research optimization solution. The result that we got fulfills all the conditions and constraints that we have declared at the starting of the problem.

The optimal profit comes in while making more burgers and the rest of the items to its minimum sale that we have specified. All the results of what quantity needs to be produced or cooked is mentioned in the output section.

The maximum profit that we can earn as a restaurant owner is Rs. 9840 per day with an investment of Rs. 20000 a day. That is a whooping 49.5 % income per day. This does not sound realistic, right? But the profit that we have got is incurred from the constraints that we have made. The constraints such as that of at least 30% of Burgers and Pizza to be sold per day and that the minimum sale requirement of every item being already set on a per day basis, forces the objective function to reach upto such a maximum.

The solution has an optimal status, that means, it is realistic given the constraints are true in the real world scenario. We analyzed the objective function by changing the RHS of the constraints and in some cases, the LHS of the constraints and found some interesting results. While a huge profit of Rs. 9840 is gained per day when the investment is Rs. 20000 per day, even if the investment is increased to Rs. 30000, there is only a minute/marginal change in the profit while in the range 20000-30000, the profit almost did not change. This is a very unrealistic scenario because in the real world, as the investment increases, the production also increases and hence the profit. But the constraints play a very pivotal role in this Linear Programming Restaurant Problem.

On changing the appliances per item per unit, there is a huge deflection in the distribution of the quantity of the item sale. In such case, the cold drinks will be sold the most up to 56-100 numbers daily to generate the same profit. A corresponding increase in profit (16000-35000) was also seen when the investment was between 30000-50000. One more rare result to notice was when the investment was changed to 80,000 keeping all the other constraints the same, there was an unbelievable profit of Rs. 85,000 which is obviously unbelievable. All these changes were reflected due to the various constraints that we have taken,

References

- Boas, Luciano Vilas. “Maximizing Profit Using Linear Programming in Python.” *Medium*, Towards Data Science, 25 June 2020, towardsdatascience.com/maximizing-profit-using-linear-programming-in-python-642520c43f6.
- Love, Robert R., and James M. Hoey. “Management Science Improves Fast-Food Operations.” *Interfaces*, vol. 20, no. 2, 1990, pp. 21–29., doi:10.1287/inte.20.2.21.

- Maiti, Arimitra. “Data Envelopment Analysis in Linear Programming Problem (LPP).” *Medium*, Towards Data Science, 23 Nov. 2020, towardsdatascience.com/data-envelopment-analysis-in-linear-programming-problem-lpp-f0f7bf57e833.

- Saci, Samir. “Maximize Your Business Profitability with Python.” *Medium*, Towards Data Science, 30 Apr. 2021, towardsdatascience.com/maximize-your-business-profitability-with-python-fbefebbdf802.

- Soni, Priyansh. “Linear Programming Using Python.” *Medium*, Towards Data Science, 27 Apr. 2020, towardsdatascience.com/linear-programming-using-python-priyansh-22b5ee888fe0.

Appendix

```
import pulp as p
prob = p.LpProblem("Maximization_of_Restaurant_Profit", p.LpMaximize)
items = {"x1": "Burger", "x2": "French Fries", "x3": "Pizza", "4": "Cold Drinks", "x5": "Pasta", "x6": "Salad", "x7": "Momos",
         "x8": "Ice Cream", "x9": "Sandwich", "x10": "Samosa"}
x1 = p.LpVariable("x1", lowBound=35, cat = 'Integer')
x2 = p.LpVariable("x2", lowBound=16, cat = 'Integer')
x3 = p.LpVariable("x3", lowBound=20, cat = 'Integer')
x4 = p.LpVariable("x4", lowBound=11, cat = 'Integer')
x5 = p.LpVariable("x5", lowBound=12, cat = 'Integer')
x6 = p.LpVariable("x6", lowBound=5, cat = 'Integer')
x7 = p.LpVariable("x7", lowBound=15, cat = 'Integer')
x8 = p.LpVariable("x8", lowBound=30, cat = 'Integer')
x9 = p.LpVariable("x9", lowBound=18, cat = 'Integer')
x10 = p.LpVariable("x10", lowBound=25, cat = 'Integer')
```

```
#Objective Function
prob += 70 * x1 + 80 * x2 + 100 * x3 + 45 * x4 + 40 * x5 + 45 * x6 + 55 * x7 + 20 * x8 + 50 * x9 + 15 * x10

#Constraints

# Max cost per item per day
prob += 150 * x1 + 90 * x2 + 250 * x3 + 95 * x4 + 110 * x5 + 50 * x6 + 75 * x7 + 80 * x8 + 40 * x9 + 20 * x10 <= 20000
# 8 workers work 12 hours per day
prob += 25 * x1 + 20 * x2 + 30 * x3 + 5 * x4 + 40 * x5 + 10 * x6 + 25 * x7 + 7 * x8 + 10 * x9 + 50 * x10 <= 5760
# Appliances and utensils used for all items per day is 350
prob += 1 * x1 + 2 * x2 + 2 * x3 + 1 * x4 + 3 * x5 + 4 * x6 + 3 * x7 + 2 * x8 + 1 * x9 + 2 * x10 <= 350
# Minimum no of sale should be 150
prob += x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 >= 150
# For a mix of a burger and pizza, people will buy 3 drinks
prob += 3 * x1 + 3 * x3 - 2 * x4 >= 0
# From total sales, 30 % sales should be at least of pizza and burger
prob += x1 + x3 >= 0.3 * (x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10)
```

```python
# print(prob)

status = prob.solve()
print(p.LpStatus[status])

x = [x1, x2, x3, x4, x5, x6, x7, x8, x9, x10]
print("The Restaurant needs to produce the items in following quantities:\n")
for i, j in zip(items, x):
    print(items[i], "   =    ", p.value(j))

print("\nAnd the maximum profit is : Rs. ", p.value(prob.objective))
```