Science Journal of Applied Mathematics and Statistics
2020; 8(5): 59-66
http://www.sciencepublishinggroup.com/j/sjams
doi: 10.11648/j.sjams.20200805.12
ISSN: 2376-9491 (Print); ISSN: 2376-9513 (Online)

Analysis of Diet Choice towards a Proper Nutrition Plan by Linear Programming

Tanzila Yeasmin Nilu[1, *], Shek Ahmed[2], Hashnayne Ahmed[2]

[1]Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh

[2]Department of Mathematics, University of Barishal, Barishal, Bangladesh

Email address:
tanzila@cse.green.edu.bd (T. Y. Nilu)

[*]Corresponding author

To cite this article:
Tanzila Yeasmin Nilu, Shek Ahmed, Hashnayne Ahmed. Analysis of Diet Choice towards a Proper Nutrition Plan by Linear Programming. *Science Journal of Applied Mathematics and Statistics.* Vol. 8, No. 5, 2020, pp. 59-66. doi: 10.11648/j.sjams.20200805.12

Received: August 9, 2020; **Accepted**: August 25, 2020; **Published**: September 21, 2020

Abstract: Linear Programming is an optimization technique to attain the most effective outcome or optimize the objective function (like maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships called the constraints. In this paper, we have discussed fundamental and detailed techniques of formulating LPs models in various real-life decision problems, decisions, works, etc. In the human body, an unhealthy diet can cause a lot of nutrition-related diseases. Sometimes, having a proper diet costs beyond one's limit and it affects us to develop a diet based budget-friendly nutrition model. Our goal is to minimize the total cost considering the required amount of nutrition values required. To construct the study we took some standard values of nutrition ingredients to compute the budget-friendly values. It's quite hard to resolve most of the real-life models with a large number of decision variables & constraints by hand calculations implies the use of AMPL (A Mathematical Programming Language) coding to get the optimal result. The number of variables & constraints isn't mattered in any respect for the computer techniques used in this study. This study results in some standard values of diet plan for optimizing the nutrition for a particular person with limited costs.

Keywords: Optimization, Linear Programming Diet, Optimization Model, Real-Life Application, AMPL, Computer-Based Program

1. Introduction

In practical life, we have to decide every step. While decision making we seek to answer the question `what is best?' Always we want the best output with limited resources. A typical example would be taking the limitations of materials and labor and then determining the "best" production levels for maximal profits under those conditions. A linear programming (LP) problem is an optimization model by which we can optimize a measure of effectiveness under conditions of allocating scarce resources and before doing that we have to formulate LP according to the given restrictions.

The problem of solving a system of linear inequalities dates back at least as far as Fourier, after whom the tactic of Fourier-Motzkin elimination is named. Linear Programming (LP) was first developed by Leonid Kantorovich in 1939 [2]. It had been used during World War II to plan expenditures and returns to cut back costs to the military and increase losses to the enemy. The three founding figures within the subject are considered to be Leonid Kantorovich, who developed the earliest LP problems in 1939, George Dantzig, who published the simplex method in 1947, and John mathematician who developed the speculation of the duality in the same year [1, 3]. The method was kept secret until 1947 when George B. Dantzig published the simplex method and John mathematician developed the idea of duality as a linear optimization solution and applied it in the field of game theory. Postwar, many industries found their use in their daily planning. The LP problem was first shown to be solvable in polynomial time by Leonid Khachiyan in 1979, but a much bigger theoretical and practical breakthrough within the field came in 1984 when Narendra Carmaker introduced a replacement idea named, the interior-point method for solving LP problems.

Dantzig's original example of finding the most effective assignment of 70 people to 70 jobs exemplifies the usefulness of linear programming [2, 4]. The computing power required to check all the permutations to pick the most effective assignment is vast the number of possible configurations exceeds the number of particles in the universe. However, it takes only a rapid to go looking out the optimum solution by posing the matter as a linear program and applying the Simplex algorithm [5]. The idea behind linear programming drastically reduces the number of possible optimal solutions that have got to be checked.

Every person needs nutrients for their sound body. A human cannot live without nutrients. The nutrient helps us to protect our body from different diseases. We can get a required amount of nutrients for our body from various kinds of foods. The amount of nutrient that is required for our body varies from age to age.

2. Preliminaries

2.1. Acquaintance with Linear Programming

Linear programming, sometimes known as linear optimization, is the problem of maximizing or minimizing a linear function over a convex polyhedron specified by linear and non-negativity constraints. Simplistically, it is the optimization of an outcome based on some set of constraints using a linear mathematical model. Optimization problems arise in all branches of Economics, Finance, Chemistry, Materials Science, Astronomy, Physics, Structural and Molecular Biology, Engineering, Computer Science, and Medicine [7-8, 26].

Linear programming is a mathematical method for determining a way to achieve the best outcome (such as maximum profit or lowest cost) in a given mathematical model for some list of requirements represented as linear relationships [7]. Linear programming is a specific case of mathematical programming (mathematical optimization).

More formally, (LP) is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Its feasible region is a convex polyhedron, which is a set defined as the intersection of finitely many half-spaces, each of which is defined by a linear inequality. Its objective function is a real-valued affine function defined on this polyhedron. A linear programming algorithm finds a point in the polyhedron where this function has the smallest (or largest) value if such a point exists [22-25, 27].

Linear programs are problems that can be expressed in canonical form:

$$Maximize\ c^T x$$

$$Subject\ to\ Ax \leq b$$

$$And\ x \geq 0$$

Where x represents the vector of variables (to be determined), c and b are vectors of (known) coefficients, A is a (known) matrix of coefficients, and $(.)^T$ is the matrix transpose. The expression to be maximized or minimized is called the objective function (cTx in this case). The inequalities $Ax \leq b$ are the constraints that specify a convex polytope over which the objective function is to be optimized.

2.2. General Form of Linear Programming

The general mathematical form of an (LP) problem is as follows:

$$\text{Optimize } Z = c_1 x_1 + c_2 x_2 + \ldots\ldots\ldots\ldots + c_n x_n$$

$$\text{Subject to: } a_{11} x_1 + a_{12} x_2 + \ldots\ldots\ldots\ldots + a_{1n} x_n \ (\leq, =, \geq)\ b_1$$

$$a_{21} x_1 + a_{22} x_2 + \ldots\ldots\ldots\ldots + a_{2n} x_n \ (\leq, =, \geq)\ b_2 \quad (1)$$

$$a_{m1} x_1 + a_{m2} x_2 + \ldots\ldots\ldots\ldots + a_{mn} x_n \ (\leq, =, \geq)\ b_m$$

$$x_j \geq 0;\ \text{ where } j = 1,2, \ldots\ldots, n.$$

Where one and only one of the signs $\leq, =, \geq$ holds for each constraint in (1) and the sign may vary from one constraint to another [5, 6]. Here $c_{j\ (j=1,2,\ldots\ldots\ldots,n)}$ are called profit (or cost) coefficients and $x_{j\ (j=1,2,\ldots\ldots\ldots,n)}$ are called decision variables.

In matrix form:

Optimize $Z = c^T x$

Subject to $Ax = b$, and $x \geq 0$.

2.3. Formulation of LP Problem

Problem formulation is the most significant part of solving LP problems. Successful optimization fully depends on the proper formulation of the problem. Formulation refers to the creating of components of the LP inappropriate mathematical relationships or structures in step with the conditions. During this section, we'll discuss how we can formulate an LP problem. The procedure for the mathematical formulation of an LP problem consists of the following major steps: [5, 14]

Step 1: To Identify Variables

We identify the unknown variables to be determined (decision variables) and represent them in terms of algebraic symbols.

Step 2: To seek out the Objective Function

We identify the objective or criterion and represent it as a linear function of the decision variables, which is to be maximized or minimized.

Step 3: To Find the Constraints

We formulate the other conditions of the problem such as resource limitations, market constraints and interrelation between variables, etc. as linear equations or inequalities in terms of decision variables.

Step 4: To Add the Non-negativity Restriction

We add the ‘Non-negativity’ constraint from the consideration that negative values of the decision variables don't have any valid physical interpretation.

Step 5: To Write Down the Entire Problem

The objective function, the set of constraints, and also the non-negative restrictions together form an LP problem.

3. Real-Life Diet Problem Analysis

3.1. Problem Definition

Every person needs nutrients for their sound body. They can get the required nutrients from various kinds of foods [10]. In this chapter, we discuss the required amount of nutrients for a person in a week in different range levels of people. We also show a linear program for the diet problem corresponding to the required amount of food and nutrients for different ages level of people.

In this project, we, work on the formulation of real-life diet problems by using the AMPL (A Mathematical Programming Language) programming [31, 26]. To establish this project paper we need so much data and information such as the nutrition value of the food, maximum and minimum required amount of nutrients for different ages, people, in a week, food cost per unit, etc. Here we have worked about 30 kinds of foods, corresponding 15 kinds of nutrients. In this project, we work on three age-levels and these are categorized as below 12 years, 12-40 years, and above 40 years. For our limitations, we have shown only the level of ages below 12 years. If any reader is interested to know the three categories you can collect the file from the authors.

We have collected the above data from various sources. Some data are collected from the internet [6, 9], some are supplied by the students of medical colleges, and the Department of Food and Nutrition. Based on per week the maximum and minimum required quantity of nutrients for each age-level and nutrition value of each food corresponding to the vitamin are collected from a book which we collect from the department of Food and Nutrition [9, 10].

Moreover, we collected the prices of these foods from the local market and converted these prices from taka into the dollar.

3.2. A Linear Programming for the Diet Problem

In this section, we will show the linear program for the real-life diet problem. To construct a linear program for the diet problem we consider the 30 foods and their corresponding 15 nutrients. For the age level below 12-years, the required amount of nutrients are given below:

The table is given below shows the maximum and minimum amount of required nutrients for a person in a week corresponding to the nutrients [11-21].

Table 1. *Minimum and maximum of required nutrients.*

parameter	n_min	n_max
CA	350	3500
CAR. HY	400	3000
CHOLES	200	3200
FTY. ACD	250	3500
FE	300	3000
K	400	7600
PRO	550	4000
NA	350	5000
A	300	4500
B12	75	1500
B6	10	1200
C	300	3500
E	100	3000
WATER	350	5000
ZN	100	5000

Here we calculate the cost of food per unit in the dollar.

Table 2. *Cost of foods per unit.*

parameter	Cost ($)	f_min	f_max
RICE	0.5	1	1.6
MILK	0.29	1	1.10
COFFEE	2	3	3.12
CALFLOWER	3	4	5.10
ORANGE	2	3	4.15
ICE	3	4	5.11
BREAD	1.09	2	3.12
OIL	1.90	2	3.19
EGG	1	2	3.10
MUSHROOMS	2.1	3	4.14
CHICKEN	1.98	3.5	4.17
BEEF	3	4	5.13
WATERMELON	3.50	5	6.16
CHILI	1.69	2	3.10
PUMPKIN	3	4	5.10
FISH	2.29	3	4.10
LITCHIS	3.5	4	5.12

parameter	Cost ($)	f_min	f_max
TURKEY	2.1	3	4.10
TOMATOES	1	2	3.10
CREACKERS	2	3	4.12
LIMEJUICE	2.79	3	4.10
PEPPER	1.26	3	4.15
GRAVELEAVES	2.1	3	3.9
GINGER	2.58	4	5.11
PEANUTS	2.78	4	5.14
LETTUCE	1.9	3	3.7
CHEESE	2.5	3	3.8
LIMES	1.22	2	3.10
NOODLES	1.39	2	3.35

The table is given below shows the number of nutrients in different kinds of food corresponding to their vitamins.

Table 3. *Amount of nutrients in different kinds of foods.*

parameter (tr)	CA	CAR. HY	CHOLES	FTY. ACD	FE	K	PRO	NA	A	B12	B6	C	E	WATER	ZN
RICE	2	21	0	0	0.1	10	2	0	0	0	0	0	0	60	0.4
MILK	80	4.7	14	2.1	0.1	52	3.3	49	26	0.4	0	0.9	0.1	88	0.4
COFFEE	2	1.5	0	0.1	0.1	80	0	14	0	0	0	0.2	0	48	0.1
CALIFLOWER	16	4.1	0	0.1	0.3	42	1.8	15	17	0	0.2	44	0	63	0.2
ORANGE	8	10	0	0	0.5	58	0.6	3	55	0	0	29	0.1	44	0.1
ICE	28	24	44	6.8	0.1	99	3.5	80	89	0.4	0	0.6	0.2	61	0.7
BREAD	55	24	1	0.6	3.3	31	9	92	0	0	0.1	0	0.3	36	0.7
OIL	0	0	0	18	0	0	0	0	0	0	0	0	28	0	0
EGG	55	1.4	59	4.2	1.6	32	14	53	85	0.9	0.1	0	1.6	69	1.2
MUSHROOMS	11	75	0	0.2	1.7	53	96	13	0	0	1	3.5	0.1	9.5	7.7
CHICKEN	14	0.9	63	1.8	8.5	40	24	51	0	19	0.6	16	1.4	68	4.3
BEEF	20	11	48	11	5.4	57	33	221	0	1	0.2	0	0.5	23	8.1
WATERMELON	54	15.3	0	9.78	7.28	64	28.3	99	0	0	0.09	0	0	5.05	10.2
CHILI	78	54.7	0	2.95	14.3	110	12.3	101	0	0	3.67	64.1	1.03	7.79	2.7
PUMPKIN	43	17.8	0	8.67	15	102	24.5	18	120	0	0.22	1.9	1	6.92	7.46
FISH	51	0	99	0.22	1.11	58	17.5	97	50	3.1	0.13	0.5	0	80.8	1.48
LITCHIS	5	16.5	0	0.1	0.31	55	0.88	1	0	0	0.1	71.5	0.7	81.8	0.07
TURKEY	25	0	76	1.64	1.78	98	29.3	70	0	0.37	0.46	0	0.33	64.9	3.1
TOMATOES	40	55.8	0	0.43	9.09	125	14.1	205	125	0	0.33	39.2	0.01	14.6	1.99
CREACKERS	84	71.5	0	2.93	5.4	89	9.2	205	0	0	0.04	0	1.55	4.1	0.77
LIMEJUICE	7	8.6	0	0	0	55	0.4	1	20	0	0.1	46	0.1	91	0.1
PEPPER	61	51	0	0.8	11	241	12	43	0	3.5	2	4.4	2.3	1.4	2.8
GRAVELEAVES	63	17	0	0.3	2.6	72	5.6	9	0	0	0.4	11	2	73	0.7
GINGER	69	71	0	1.9	12	343	9.1	32	47	0	0.8	7	0.3	9.4	4.7
PEANUTS	54	22	0	6.9	2.3	65	24	6	0	0	0.9	0	7.8	1.6	3.3
LETTUCE	32	2.3	0	0	0.3	57	1.3	5	47	0	0.7	8	0.4	96	0.2
CHEESE	88	0.5	72	15	0.3	87	20	84	23	1.3	0.2	0	0.7	52	24
LIMES	33	11	0	0	0.6	102	0.7	2	10	0	0	29	0.2	88	0.1
NOODLES	20	58	0	4.4	4.7	120	8.4	96	85	0	0.1	0	0.2	0.7	14

Now we can construct a linear program according to the given system is,

$$Minimize\ Z = \sum_{j=1}^{n} c_j x_j$$

$$Subject\ to, \sum_{j}^{n} a_{ij}\, x_j \geq b_i\ where\ i = 1,2, \dots\dots\dots, m$$

$$and\ x_j \geq 0$$

where,

x_j: The food j has eaten per week

c_j: The price of food j per $1000g$

a_{ij}: The amount of nutrient i in 1000g of food j

b_i: The required weekly amount of nutrient i

m: The number of nutrients

n: The number of food

Now we can construct a linear program by using AMPL [28-32].

Minimize Total_Cost:

0.5*Buy[‘RICE’] + 0.29*Buy[‘MILK’] + 2*Buy[‘COFFEE’] + 3*Buy[‘CALIFLOWER’] + 2*Buy[‘ORANGE’] + 3*Buy[‘ICE’] + 1.09*Buy[‘BREAD’] + 1.9*Buy[‘OIL’] +Buy[‘EGG’] + 2.1*Buy[‘MUSHROOMS’] + 1.98*Buy[‘CHICKEN’] +

3*Buy['BEEF'] + 3.5*Buy['WATERMELON'] + 1.69*Buy['CHILI'] + 3*Buy['PUMPKIN'] + 2.29*Buy['FISH'] + 3.5*Buy['LITCHIS'] + 2.1*Buy['TURKEY'] + Buy['TOMATOES'] + 2*Buy['CREACKERS'] + 2.79*Buy['LIMEJUICE'] + 1.26*Buy['PEPPER'] + 2.1*Buy['GRAVELEAVES'] + 2.58*Buy['GINGER'] +2.78*Buy['PEANUTS'] + 1.9*Buy['LETTUCE'] + 2.5*Buy['CHEESE'] + 1.22*Buy['LIMES'] + 1.39*Buy['NOODLES'];

Subject to Diet['CA']:

350<=2*Buy['RICE'] + 80*Buy['MILK'] + 2*Buy['COFFEE'] + 16*Buy['CALIFLOWER'] + 8*Buy['ORANGE'] + 28*Buy['ICE'] + 55*Buy['BREAD'] + 0*Buy['OIL'] + 55*Buy['EGG'] + 11*Buy['MUSHROOMS'] + 14*Buy['CHICKEN'] + 20*Buy['BEEF'] + 54*Buy['WATERMELON'] + 78*Buy['CHILI'] + 43*Buy['PUMPKIN'] + 51*Buy['FISH'] + 5*Buy['LITCHIS'] + 25*Buy['TURKEY'] + 40*Buy['TOMATOES'] + 84*Buy['CREACKERS'] + 7*Buy['LIMEJUICE'] + 61*Buy['PEPPER'] + 63*Buy['GRAVELEAVES'] + 69*Buy['GINGER'] + 54*Buy['PEANUTS'] + 32*Buy['LETTUCE'] + 88*Buy['CHEESE'] + 33*Buy['LIMES'] + 20*Buy['NOODLES'] <=3500;

Subject to Diet['CAR.HY']:

400<=21*Buy['RICE'] + 4.7*Buy['MILK'] + 1.5*Buy['COFFEE'] + 4.1*Buy['CALIFLOWER'] + 10*Buy['ORANGE'] + 24*Buy['ICE'] + 24*Buy['BREAD'] + 0*Buy['OIL'] + 1.4*Buy['EGG'] + 75*Buy['MUSHROOMS'] + 0.9*Buy['CHICKEN'] + 11*Buy['BEEF'] + 15.3*Buy['WATERMELON'] + 54.7*Buy['CHILI'] + 17.8*Buy['PUMPKIN'] + 0*Buy['FISH'] + 16.5*Buy['LITCHIS'] + 0*Buy['TURKEY'] + 55.8*Buy['TOMATOES'] + 71.5*Buy['CREACKERS'] + 8.6*Buy['LIMEJUICE'] + 51*Buy['PEPPER'] + 17*Buy['GRAVELEAVES'] + 71*Buy['GINGER'] + 22*Buy['PEANUTS'] + 2.3*Buy['LETTUCE'] + 0.5*Buy['CHEESE'] + 11*Buy['LIMES'] + 58*Buy['NOODLES'] <=3000;

Subject to Diet['CHOLES']:

0*Buy['LIMEJUICE'] + 11*Buy['PEPPER'] + 2.6*Buy['GRAVELEAVES'] + 12*Buy['GINGER'] + 2.3*Buy['PEANUTS'] + 0.3*Buy['LETTUCE'] + 0.3*Buy['CHEESE'] + 0.6*Buy['LIMES'] + 4.7*Buy['NOODLES'] <=3000;

Subject to Diet['K']:

400<=10*Buy['RICE'] + 52*Buy['MILK'] + 80*Buy['COFFEE'] + 42*Buy['CALIFLOWER'] + 58*Buy['ORANGE'] + 99*Buy['ICE'] + 31*Buy['BREAD'] + 0*Buy['OIL'] + 32*Buy['EGG'] + 53*Buy['MUSHROOMS'] + 40*Buy['CHICKEN'] + 57*Buy['BEEF'] + 64*Buy['WATERMELON'] + 110*Buy['CHILI'] + 102*Buy['PUMPKIN'] + 58*Buy['FISH'] + 55*Buy['LITCHIS'] + 98*Buy['TURKEY'] + 125*Buy['TOMATOES'] + 89*Buy['CREACKERS'] + 55*Buy['LIMEJUICE'] + 241*Buy['PEPPER'] + 72*Buy['GRAVELEAVES'] +

200<=0*Buy['RICE'] + 14*Buy['MILK'] + 0*Buy['COFFEE'] + 0*Buy['CALIFLOWER'] + 0*Buy['ORANGE'] + 44*Buy['ICE'] + 1*Buy['BREAD'] + 0*Buy['OIL'] + 59*Buy['EGG'] + 0*Buy['MUSHROOMS'] + 63*Buy['CHICKEN'] + 48*Buy['BEEF'] + 0*Buy['WATERMELON'] + 0*Buy['CHILI'] + 0*Buy['PUMPKIN'] + 99*Buy['FISH'] + 0*Buy['LITCHIS'] + 76*Buy['TURKEY'] + 0*Buy['TOMATOES'] + 0*Buy['CREACKERS'] + 0*Buy['LIMEJUICE'] + 0*Buy['PEPPER'] + 0*Buy['GRAVELEAVES'] + 0*Buy['GINGER'] + 0*Buy['PEANUTS'] + 0*Buy['LETTUCE'] + 72*Buy['CHEESE'] + 0*Buy['LIMES'] + 0*Buy['NOODLES'] <=3200;

Subject to Diet['FTY.ACD']:

250<=0*Buy['RICE'] + 2.1*Buy['MILK'] + 0.1*Buy['COFFEE'] + 0.1*Buy['CALIFLOWER'] + 0*Buy['ORANGE'] + 6.8*Buy['ICE'] + 0.6*Buy['BREAD'] + 18*Buy['OIL'] + 4.2*Buy['EGG'] + 0.2*Buy['MUSHROOMS'] + 1.8*Buy['CHICKEN'] + 11*Buy['BEEF'] + 9.78*Buy['WATERMELON'] + 2.95*Buy['CHILI'] + 8.67*Buy['PUMPKIN'] + 0.22*Buy['FISH'] + 0.1*Buy['LITCHIS'] + 1.64*Buy['TURKEY'] + 0.43*Buy['TOMATOES'] + 2.93*Buy['CREACKERS'] + 0*Buy['LIMEJUICE'] + 0.8*Buy['PEPPER'] + 0.3*Buy['GRAVELEAVES'] + 1.9*Buy['GINGER'] + 6.9*Buy['PEANUTS'] + 0*Buy['LETTUCE'] + 15*Buy['CHEESE'] + 0*Buy['LIMES'] + 4.4*Buy['NOODLES'] <=3500;

Subject to Diet['FE']:

300<=0.1*Buy['RICE'] + 0.1*Buy['MILK'] + 0.1*Buy['COFFEE'] + 0.3*Buy['CALIFLOWER'] + 0.5*Buy['ORANGE'] + 0.1*Buy['ICE'] + 3.3*Buy['BREAD'] + 0*Buy['OIL'] + 1.6*Buy['EGG'] + 1.7*Buy['MUSHROOMS'] + 8.5*Buy['CHICKEN'] + 5.4*Buy['BEEF'] + 7.28*Buy['WATERMELON'] + 14.3*Buy['CHILI'] + 15*Buy['PUMPKIN'] + 1.11*Buy['FISH'] + 0.31*Buy['LITCHIS'] + 1.78*Buy['TURKEY'] + 9.09*Buy['TOMATOES'] + 5.4*Buy['CREACKERS'] + 343*Buy['GINGER'] + 65*Buy['PEANUTS'] + 57*Buy['LETTUCE'] + 87*Buy['CHEESE'] + 102*Buy['LIMES'] + 120*Buy['NOODLES'] <=7600;

Subject to Diet['PRO']:

550<=2*Buy['RICE'] + 3.3*Buy['MILK'] + 0*Buy['COFFEE'] + 1.8*Buy['CALIFLOWER'] + 0.6*Buy['ORANGE'] + 3.5*Buy['ICE'] + 9*Buy['BREAD'] + 0*Buy['OIL'] + 14*Buy['EGG'] + 96*Buy['MUSHROOMS'] + 24*Buy['CHICKEN'] + 33*Buy['BEEF'] + 28.3*Buy['WATERMELON'] + 12.3*Buy['CHILI'] + 24.5*Buy['PUMPKIN'] + 17.5*Buy['FISH'] + 0.88*Buy['LITCHIS'] + 29.3*Buy['TURKEY'] + 14.1*Buy['TOMATOES'] + 9.2*Buy['CREACKERS'] + 0.4*Buy['LIMEJUICE'] + 12*Buy['PEPPER'] + 5.6*Buy['GRAVELEAVES'] + 9.1*Buy['GINGER'] + 24*Buy['PEANUTS'] + 1.3*Buy['LETTUCE'] + 20*Buy['CHEESE'] +

0.7*Buy[‘LIMES’] + 8.4*Buy[‘NOODLES’] <=4000;

Subject to Diet[‘NA’]:

350<=0*Buy[‘RICE’] + 49*Buy[‘MILK’] + 14*Buy[‘COFFEE’] + 15*Buy[‘CALIFLOWER’] + 3*Buy[‘ORANGE’] + 80*Buy[‘ICE’] + 92*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 53*Buy[‘EGG’] + 13*Buy[‘MUSHROOMS’] + 51*Buy[‘CHICKEN’] + 221*Buy[‘BEEF’] + 99*Buy[‘WATERMELON’] + 101*Buy[‘CHILI’] + 18*Buy[‘PUMPKIN’] + 97*Buy[‘FISH’] + 1*Buy[‘LITCHIS’] + 70*Buy[‘TURKEY’] + 205*Buy[‘TOMATOES’] + 205*Buy[‘CREACKERS’] + 1*Buy[‘LIMEJUICE’] + 43*Buy[‘PEPPER’] + 9*Buy[‘GRAVELEAVES’] + 32*Buy[‘GINGER’] + 6*Buy[‘PEANUTS’] + 5*Buy[‘LETTUCE’] + 84*Buy[‘CHEESE’] + 2*Buy[‘LIMES’] + 96*Buy[‘NOODLES’] <=5000;

Subject to Diet[‘A’]:

300<=0*Buy[‘RICE’] + 26*Buy[‘MILK’] + 0*Buy[‘COFFEE’] + 17*Buy[‘CALIFLOWER’] + 55*Buy[‘ORANGE’] + 89*Buy[‘ICE’] + 0*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 85*Buy[‘EGG’] + 0*Buy[‘MUSHROOMS’] + 0*Buy[‘CHICKEN’] + 0*Buy[‘BEEF’] + 0*Buy[‘WATERMELON’] + 0*Buy[‘CHILI’] + 120*Buy[‘PUMPKIN’] + 50*Buy[‘FISH’] + 0*Buy[‘LITCHIS’] + 0*Buy[‘TURKEY’] + 125*Buy[‘TOMATOES’] + 0*Buy[‘CREACKERS’] + 20*Buy[‘LIMEJUICE’] + 0*Buy[‘PEPPER’] + 0*Buy[‘GRAVELEAVES’] + 47*Buy[‘GINGER’] + 0*Buy[‘PEANUTS’] + 47*Buy[‘LETTUCE’] + 23*Buy[‘CHEESE’] + 10*Buy[‘LIMES’] + 85*Buy[‘NOODLES’] <=4500;

Subject to Diet[‘B12’]:

75<=0*Buy[‘RICE’] + 0.4*Buy[‘MILK’] + 0*Buy[‘COFFEE’] + 0*Buy[‘CALIFLOWER’] + 0*Buy[‘ORANGE’] + 0.4*Buy[‘ICE’] + 0*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 0.9*Buy[‘EGG’] + 0*Buy[‘MUSHROOMS’] + 19*Buy[‘CHICKEN’] + 1*Buy[‘BEEF’] + 0*Buy[‘WATERMELON’] + 0*Buy[‘CHILI’] + 0*Buy[‘PUMPKIN’] + 3.1*Buy[‘FISH’] + 0*Buy[‘LITCHIS’] + 0.37*Buy[‘TURKEY’] + 0*Buy[‘TOMATOES’] + 0*Buy[‘CREACKERS’] + 0*Buy[‘LIMEJUICE’] + 3.5*Buy[‘PEPPER’] + 0*Buy[‘GRAVELEAVES’] + 0*Buy[‘GINGER’] + 0*Buy[‘PEANUTS’] + 0*Buy[‘LETTUCE’] + 1.3*Buy[‘CHEESE’] + 0*Buy[‘LIMES’] + 0*Buy[‘NOODLES’] <=1500;

Subject to Diet[‘B6’]:

10<=0*Buy[‘RICE’] + 0*Buy[‘MILK’] + 0*Buy[‘COFFEE’] + 0.2*Buy[‘CALIFLOWER’] + 0*Buy[‘ORANGE’] + 0*Buy[‘ICE’] + 0.1*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 0.1*Buy[‘EGG’] + 1*Buy[‘MUSHROOMS’] + 0.6*Buy[‘CHICKEN’] + 0.2*Buy[‘BEEF’] + 0.09*Buy[‘WATERMELON’] + 3.67*Buy[‘CHILI’] + 0.22*Buy[‘PUMPKIN’] + 0.13*Buy[‘FISH’] + 0.1*Buy[‘LITCHIS’] + 0.46*Buy[‘TURKEY’] + 0.33*Buy[‘TOMATOES’] + 0.04*Buy[‘CREACKERS’] + 0.1*Buy[‘LIMEJUICE’] + 2*Buy[‘PEPPER’] + 0.4*Buy[‘GRAVELEAVES’] + 0.8*Buy[‘GINGER’] + 0.9*Buy[‘PEANUTS’] + 0.7*Buy[‘LETTUCE’] + 0.2*Buy[‘CHEESE’] + 0*Buy[‘LIMES’] + 0.1*Buy[‘NOODLES’] <=1200;

Subject to Diet[‘C’]:

300<=0*Buy[‘RICE’] + 0.9*Buy[‘MILK’] + 0.2*Buy[‘COFFEE’] + 44*Buy[‘CALIFLOWER’] + 29*Buy[‘ORANGE’] + 0.6*Buy[‘ICE’] + 0*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 0*Buy[‘EGG’] + 3.5*Buy[‘MUSHROOMS’] + 16*Buy[‘CHICKEN’] + 0*Buy[‘BEEF’] + 0*Buy[‘WATERMELON’] + 64.1*Buy[‘CHILI’] + 1.9*Buy[‘PUMPKIN’] + 0.5*Buy[‘FISH’] + 71.5*Buy[‘LITCHIS’] + 0*Buy[‘TURKEY’] + 39.2*Buy[‘TOMATOES’] + 0*Buy[‘CREACKERS’] + 46*Buy[‘LIMEJUICE’] + 4.4*Buy[‘PEPPER’] + 11*Buy[‘GRAVELEAVES’] + 7*Buy[‘GINGER’] + 0*Buy[‘PEANUTS’] + 8*Buy[‘LETTUCE’] + 0*Buy[‘CHEESE’] + 29*Buy[‘LIMES’] + 0*Buy[‘NOODLES’] <=3500;

Subject to Diet[‘E’]:

100<=0*Buy[‘RICE’] + 0.1*Buy[‘MILK’] + 0*Buy[‘COFFEE’] + 0*Buy[‘CALIFLOWER’] + 0.1*Buy[‘ORANGE’] + 0.2*Buy[‘ICE’] + 0.3*Buy[‘BREAD’] + 28*Buy[‘OIL’] + 1.6*Buy[‘EGG’] + 0.1*Buy[‘MUSHROOMS’] + 1.4*Buy[‘CHICKEN’] + 0.5*Buy[‘BEEF’] + 0*Buy[‘WATERMELON’] + 1.03*Buy[‘CHILI’] + 1*Buy[‘PUMPKIN’] + 0*Buy[‘FISH’] + 0.7*Buy[‘LITCHIS’] + 0.33*Buy[‘TURKEY’] + 0.01*Buy[‘TOMATOES’] + 1.55*Buy[‘CREACKERS’] + 0.1*Buy[‘LIMEJUICE’] + 2.3*Buy[‘PEPPER’] + 2*Buy[‘GRAVELEAVES’] + 0.3*Buy[‘GINGER’] + 7.8*Buy[‘PEANUTS’] + 0.4*Buy[‘LETTUCE’] + 0.7*Buy[‘CHEESE’] + 0.2*Buy[‘LIMES’] + 0.2*Buy[‘NOODLES’] <=3000;

Subject to Diet[‘WATER’]:

350<=60*Buy[‘RICE’] + 88*Buy[‘MILK’] + 48*Buy[‘COFFEE’] + 63*Buy[‘CALIFLOWER’] + 44*Buy[‘ORANGE’] + 61*Buy[‘ICE’] + 36*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 69*Buy[‘EGG’] + 9.5*Buy[‘MUSHROOMS’] + 68*Buy[‘CHICKEN’] + 23*Buy[‘BEEF’] + 5.05*Buy[‘WATERMELON’] + 7.79*Buy[‘CHILI’] + 6.92*Buy[‘PUMPKIN’] + 80.8*Buy[‘FISH’] + 81.8*Buy[‘LITCHIS’] + 64.9*Buy[‘TURKEY’] + 14.6*Buy[‘TOMATOES’] + 4.1*Buy[‘CREACKERS’] + 9.1*Buy[‘LIMEJUICE’] + 1.4*Buy[‘PEPPER’] + 73*Buy[‘GRAVELEAVES’] + 9.4*Buy[‘GINGER’] + 1.6*Buy[‘PEANUTS’] + 96*Buy[‘LETTUCE’] + 52*Buy[‘CHEESE’] + 88*Buy[‘LIMES’] + 0.7*Buy[‘NOODLES’] <=5000;

Subject to Diet[‘ZN’]:

100<=0.4*Buy[‘RICE’] + 0.4*Buy[‘MILK’] + 0.1*Buy[‘COFFEE’] + 0.2*Buy[‘CALIFLOWER’] + 0.1*Buy[‘ORANGE’] + 0.7*Buy[‘ICE’] + 0.7*Buy[‘BREAD’] + 0*Buy[‘OIL’] + 1.2*Buy[‘EGG’] + 7.7*Buy[‘MUSHROOMS’] + 4.3*Buy[‘CHICKEN’] + 8.1*Buy[‘BEEF’] + 10.2*Buy[‘WATERMELON’] + 2.7*Buy[‘CHILI’] + 7.46*Buy[‘PUMPKIN’] +

1.48*Buy['FISH'] + 0.07*Buy['LITCHIS'] + 3.1*Buy['TURKEY'] + 1.99*Buy['TOMATOES'] + 0.77*Buy['CREACKERS'] + 0.1*Buy['LIMEJUICE'] + 2.8*Buy['PEPPER'] + 0.7*Buy['GRAVELEAVES'] + 4.7*Buy['GINGER'] + 3.3*Buy['PEANUTS'] + 0.2*Buy['LETTUCE'] + 24*Buy['CHEESE'] + 0.1*Buy['LIMES'] + 14*Buy['NOODLES'] <=5000;

In the same process, we can construct a linear program for the different age levels of people.

3.3. Computer-Based Solution Techniques

AMPL Program

(Input model file)

```
set NUTR;
set FOOD;
param cost {FOOD} > 0;
param f_min {FOOD}>=0;
param f_max {j in FOOD} >=f_min [j];
param n_min {NUTR} >=0;
param n_max {I in NUTR} >=n_min [i];
param amt {NUTR, FOOD} >=0;
var Buy {j in FOOD} >=f_min[j], <=f_max [j];
minimize Total_Cost: sum {j in FOOD} cost[j]*Buy[j];
subject to Diet {i in NUTR}:
n_min[i] <=sum {j in FOOD} amt[i, j] * Buy[j] <=n_max[i];
```

(Input data file)

```
Set NUTR:=CA CAR.HY CHOLES FTY.ACD FE K PRO NA A B12 B6 C E WATER ZN;
set FOOD:=RICE MILK COFFEE CAULIFLOWER ORANGE ICE BREAD OIL EGG MUSHROOMS CHICKEN BEEF WATERMELON CHILI PUMPKIN FISH LITCHIS TURKEY TOMATOES CRACKERS LIMEJUICE PEPPER GRAVELEAVES GINGER PEANUTS LETTUCE CHEESE LIMES NOODLES;
```

(Output file)

```
sw: ampl
ampl: model d1mod.txt;
ampl: data p1dat.txt;
ampl: solve;
MINOS 5.5: optimal solution found.
13 iterations, objective 196.36
ampl: display Buy;
Buy [*]:=
```

BEEF	4	EGG	2	LITCHIS	4	PUMPKIN	4
BREAD	2	FISH	3	MILK	1	RICE	1
CAULIFLOWER	4	GINGER	4	MUSHROOMS	3	TOMATOES	2
CHEESE	3	GRAVELEAVES	3	NOODLES	2	TURKEY	3
CHICKEN	3.5	ICE	4	OIL	2	WATERMELON	5
CHILI	2	LETTUCE	3	ORANGE	3		
COFFEE	3	LIMEJUICE	3	PEANUTS	4		
CREACKERS	3	LIMES	2	PEPPER	3		

Thus we see that if they eat 4 units of BEEF, 2 units of BREAD, 4 units of CAULIFLOWER, 3 units of CHEESE, 3.5 units CHICKEN, 2 units of CHILI, 3 units COFFEE, 3 units of CRACKERS, 2 units of the EGG, 3 units of FISH, 4 units of GINGER, 3 units of GRAVELEAVES, 4 units of ICE, 3 units of LETTUCE, 2 units of LIMES, 4 units LITCHIS, 1 unit of MILK, 3 units of MUSHROOMS, 2 units of NOODLES, 2 units of OIL, 3 units of ORANGE, 4 units of PEANUTS, 3 units of PEPPER, 4 units of PUMPKIN, 1 unit of RICE, 2 units of TOMATOES, 3 units of TURKEY and 5 units of WATERMELON, they can get the maximum amount of nutrient foods as well as minimize their total cost.

4. Conclusion

In this paper, we have presented some useful techniques to formulate a large scale Linear Programming problem by considering the restrictions of materials, labor, etc. We also described a real-life model of a diet problem to make the best decisions and analyzed the model considering as a Linear Optimization problem. Anyone can lose profit in his business without gathering vast knowledge in the LP problem, but if he can take decisions by ascertaining his deficiency of sources can get the best output. Most often, it is very difficult to compute the profit manually, so we used a coding system AMPL to attenuate time-consuming.

Credit Authorship Contribution Statement

Tanzila Yeasmin Nilu: Conceptualization, Methodology, Investigation, Data collection, Formal Analysis, Validation, Writing-original draft. Shek Ahmed: Conceptualization, Methodology, Investigation, Formal Analysis, Validation, Writing-original draft. Hashnayne Ahmed: Writing-original draft, Writing-review & editing.

Declaration of Competing Interest

The authors declare that they have no competing interests.

Role of Funding Source

Self-funded.

Ethical Approval

Not required.

References

[1] Hillier, Frederick S. *Introduction to operations research*. Tata McGraw-Hill Education, 2012.

[2] Kantorovich, L. V. "On one effective method of solving some classes of extreme problems." *Reports from the Academy of Sciences*. Vol. 28. No. 3. Federal State Unitary Enterprise Academic Science, Publishing, Printing and Book Distribution Center Science, 1940.

[3] Stapel, Elizabeth. "Linear Programming: Introduction." Purple math. Available from https://www.purplemath.com/modules/linprog.htm

[4] Noyes, James, and Weisstein, Eric W. "Linear Programming." From MathWorld-A Wolfram Web Resource. https://mathworld.wolfram.com/LinearProgramming.html.

[5] Rezaul Karim, et. al. "Linear Programming".

[6] Topcu, Y. İlker. "Operations Research II Lecture Notes." (2016).

[7] Saigal, Romesh. "Linear programming: a modern integrated analysis." *Journal of the Operational Research Society* 48.7 (1997): 762-762.

[8] Dantzig, George B., and Mukund N. Thapa. *Linear Programming. 1, Introduction {Springer Series in Operations Research}*. Springer-Verlag New York Incorporated, 1997.

[9] Dent, John Barry, and Harold Casey. "Linear programming and animal nutrition." *Linear programming and animal nutrition*. (1967).

[10] Ferguson, Elaine L., et al. "Food-based dietary guidelines can be developed and tested using linear programming analysis." *The Journal of Nutrition* 134.4 (2004): 951-957.

[11] Darmon, Nicole, Elaine L. Ferguson, and André Briend. "A cost constraint alone has adverse effects on food selection and nutrient density: an analysis of human diets by linear programming." *The Journal of Nutrition* 132.12 (2002): 3764-3771.

[12] Santika, Otte, Umi Fahmida, and Elaine L. Ferguson. "Development of food-based complementary feeding recommendations for 9-to 11-month-old peri-urban Indonesian infants using linear programming." *The Journal of Nutrition* 139.1 (2009): 135-141.

[13] Calkins, Peter H. "Nutritional adaptations of linear programming for planning rural development." *American Journal of Agricultural Economics* 63.2 (1981): 247-254.

[14] Fletcher, L. R., P. M. Soden, and A. S. I. Zinober. "Linear programming techniques for the construction of palatable human diets." *Journal of the Operational Research Society* 45.5 (1994): 489-496.

[15] Foytik, Jerry. "Very low-cost nutritious diet plans designed by linear programming." *Journal of Nutrition Education* 13.2 (1981): 63-66.

[16] Eghbali, Hossein. "Implement a Proper Nutrition Model for Athletes Using the Linear Optimization Model." *Available at SSRN 3568367* (2020).

[17] Ibrahim, Nur Fadhilah, Thurga Dewi Arunasalan, and Nurul Akmal Mohamed. "Cheap vs healthy: Analyzing McDonald's menu using linear programming." *Bulletin of Electrical Engineering and Informatics* 9.2 (2020): 771-776.

[18] Mallick, Priyaranjan, et al. "Broiler Poultry Feed Cost Optimization Using Linear Programming Technique." *Journal of Operations and Strategic Planning* (2020): 2516600X19896910.

[19] Very-Jr, Eliseu, et al. "Planning dietary improvements without additional costs for low-income individuals in Brazil: linear programming optimization as a tool for public policy in nutrition and health." *Nutrition Journal* 18.1 (2019): 40.

[20] Alaini, Reham, Roslee Rajikan, and Siti Masitah Elias. "Diet optimization using linear programming to develop low-cost cancer prevention food plan for selected adults in Kuala Lumpur, Malaysia." *BMC public health* 19.4 (2019): 546.

[21] Henson, Spencer. "Linear programming analysis of constraints upon human diets." *Journal of Agricultural Economics* 42.3 (1991): 380-393.

[22] Mansini, Renata, Wlodzimierz Ogryczak, and M. Grazia Speranza. "Twenty years of linear programming based portfolio optimization." *European Journal of Operational Research* 234.2 (2014): 518-535.

[23] Kabak, Özgür, and Füsun Ülengin. "Possibilistic linear-programming approach for supply chain networking decisions." *European Journal of Operational Research* 209.3 (2011): 253-264.

[24] Chretienne, Philippe, et al. "Scheduling theory and its applications." *Journal of the Operational Research Society* 48.7 (1997): 764-765.

[25] Ben-Tal, Aharon, and Arkadi Nemirovski. "Robust solutions of uncertain linear programs." *Operations research letters* 25.1 (1999): 1-13.

[26] Bhuiyan, Md Anwarul Islam, and Shek Ahmed. "A new computer-oriented technique for solving a linear programming problem using bender's decomposition method."

[27] Nilu, Tanzila Yeasmin, Shek Ahmed, and M. A. I. Bhuiyan. "A study of sensitivity analysis in linear programming problem and its implementation in real life."

[28] Fourer, Robert, David M. Gay, and Brian W. Kernighan. *AMPL: A mathematical programming language*. Murray Hill, NJ: AT & T Bell Laboratories, 1987.

[29] Ahmed, Hashnayne, and Shek Ahmed. "A Comparative Study on Harvesting Plan Predicting Insurance with Two-Stage Stochastic Analysis." *International Journal on Data Science and Technology* 5.4 (2019): 73.

[30] Fourer, Robert. "AMPL models for not linear optimization using linear solvers." *Proc. EURO-INFORMS Joint Int. Meeting, Rome, Italy*. 2013.

[31] Hsu, Chien-Ning, et al. "The interplay between a maternal and post-weaning high-fat diet and gut microbiota in the developmental programming of hypertension." *Nutrients* 11.9 (2019): 1982.

[32] Gay, David M. "The AMPL modeling language: An aid to formulating and solving optimization problems." *Numerical analysis and optimization*. Springer, Cham, 2015. 95-116.