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

Tanzila Yeasmin Nilu¹,*, Shek Ahmed², Hashnayne Ahmed²

¹Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh
²Department of Mathematics, University of Barishal, Barishal, Bangladesh

Email address:
tanze@cse.green.edu.bd (T. Y. Nilu)
*Corresponding author

To cite this article:

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 quick lookout 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. Simplicially, 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 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 \((c)^T \) is the matrix transpose. The expression to be maximized or minimized is called the objective function (\( c^T x \) 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:

Optimize \( Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \)

Subject to:

\[ a_{ij} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \leq b_j \]

\[ a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = b_j \]

\[ a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq b_j \]

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 \( c_j (j=1,2,\ldots,n) \) are called profit (or cost) coefficients and \( x_j (j=1,2,\ldots,n) \) 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:

<table>
<thead>
<tr>
<th>parameter</th>
<th>n_min</th>
<th>n_max</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>350</td>
<td>3500</td>
</tr>
<tr>
<td>CAR. HY</td>
<td>400</td>
<td>3000</td>
</tr>
<tr>
<td>CHOLESTEROL</td>
<td>200</td>
<td>3200</td>
</tr>
<tr>
<td>FTY. ACID</td>
<td>250</td>
<td>3500</td>
</tr>
<tr>
<td>FE</td>
<td>300</td>
<td>3000</td>
</tr>
<tr>
<td>K</td>
<td>400</td>
<td>7600</td>
</tr>
<tr>
<td>PRO</td>
<td>550</td>
<td>4000</td>
</tr>
<tr>
<td>NA</td>
<td>350</td>
<td>5000</td>
</tr>
<tr>
<td>A</td>
<td>300</td>
<td>4500</td>
</tr>
<tr>
<td>B12</td>
<td>75</td>
<td>1500</td>
</tr>
<tr>
<td>B6</td>
<td>10</td>
<td>1200</td>
</tr>
<tr>
<td>C</td>
<td>300</td>
<td>3500</td>
</tr>
<tr>
<td>E</td>
<td>100</td>
<td>3000</td>
</tr>
<tr>
<td>WATER</td>
<td>350</td>
<td>5000</td>
</tr>
<tr>
<td>ZN</td>
<td>100</td>
<td>5000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>parameter</th>
<th>Cost ($)</th>
<th>f_min</th>
<th>f_max</th>
</tr>
</thead>
<tbody>
<tr>
<td>RICE</td>
<td>0.5</td>
<td>1</td>
<td>1.6</td>
</tr>
<tr>
<td>MILK</td>
<td>0.29</td>
<td>1</td>
<td>1.10</td>
</tr>
<tr>
<td>COFFEE</td>
<td>2</td>
<td>3</td>
<td>3.12</td>
</tr>
<tr>
<td>CALFLOWER</td>
<td>3</td>
<td>4</td>
<td>5.10</td>
</tr>
<tr>
<td>ORANGE</td>
<td>2</td>
<td>3</td>
<td>4.15</td>
</tr>
<tr>
<td>ICE</td>
<td>3</td>
<td>4</td>
<td>5.11</td>
</tr>
<tr>
<td>BREAD</td>
<td>1.09</td>
<td>2</td>
<td>3.12</td>
</tr>
<tr>
<td>OIL</td>
<td>1.90</td>
<td>2</td>
<td>3.19</td>
</tr>
<tr>
<td>EGG</td>
<td>1</td>
<td>2</td>
<td>3.10</td>
</tr>
<tr>
<td>MUSHROOMS</td>
<td>2.1</td>
<td>3</td>
<td>4.14</td>
</tr>
<tr>
<td>CHICKEN</td>
<td>1.98</td>
<td>3.5</td>
<td>4.17</td>
</tr>
<tr>
<td>BEEF</td>
<td>3</td>
<td>4</td>
<td>5.13</td>
</tr>
<tr>
<td>WATERMELON</td>
<td>3.50</td>
<td>5</td>
<td>6.16</td>
</tr>
<tr>
<td>CHILI</td>
<td>1.69</td>
<td>2</td>
<td>3.10</td>
</tr>
<tr>
<td>PUMPKIN</td>
<td>3</td>
<td>4</td>
<td>5.10</td>
</tr>
<tr>
<td>FISH</td>
<td>2.29</td>
<td>3</td>
<td>4.10</td>
</tr>
<tr>
<td>LITCHIS</td>
<td>3.5</td>
<td>4</td>
<td>5.12</td>
</tr>
</tbody>
</table>
The table is given below shows the number of nutrients in different kinds of food corresponding to their vitamins.

<table>
<thead>
<tr>
<th>parameter</th>
<th>Cost ($)</th>
<th>$f_{min}$</th>
<th>$f_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TURKEY</td>
<td>2.1</td>
<td>3</td>
<td>4.10</td>
</tr>
<tr>
<td>TOMATOES</td>
<td>1</td>
<td>2</td>
<td>3.10</td>
</tr>
<tr>
<td>CREACKERS</td>
<td>2</td>
<td>3</td>
<td>4.12</td>
</tr>
<tr>
<td>LIMEJUICE</td>
<td>2.79</td>
<td>3</td>
<td>4.10</td>
</tr>
<tr>
<td>PEPPER</td>
<td>1.26</td>
<td>3</td>
<td>4.15</td>
</tr>
<tr>
<td>GRAVELEAVES</td>
<td>2.1</td>
<td>3</td>
<td>3.9</td>
</tr>
<tr>
<td>GINGER</td>
<td>2.58</td>
<td>4</td>
<td>5.11</td>
</tr>
<tr>
<td>PEANUTS</td>
<td>2.78</td>
<td>4</td>
<td>5.14</td>
</tr>
<tr>
<td>LETTUCE</td>
<td>1.9</td>
<td>3</td>
<td>3.7</td>
</tr>
<tr>
<td>CHEESE</td>
<td>2.5</td>
<td>3</td>
<td>3.8</td>
</tr>
<tr>
<td>LIMES</td>
<td>1.22</td>
<td>2</td>
<td>3.10</td>
</tr>
<tr>
<td>NOODLES</td>
<td>1.39</td>
<td>2</td>
<td>3.35</td>
</tr>
</tbody>
</table>

The price of food $j$ per 1000g

$A_{ij}$: The amount of nutrient $i$ in $1000$g 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 according to the given system is,

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

Subject to,

$$
\sum_{j} a_{ij} x_j \geq b_i \text{ where } i = 1, 2, \ldots, m
$$

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"] + 1.9*Buy["EGG"] + 2.1*Buy["MUSHROOMS"] + 1.98*Buy["CHICKEN"] + ...
Subject to Diet ['CA']:

350 <= 2*Buy['RICE'] + 80*Buy['MILK'] + 10*Buy['COFFEE'] + 34*Buy['CHICKEN'] + 11*Buy['LETTUCE'] + 33*Buy['NOODLES'] <= 3500;

subject to Diet ['CHOLES']:

400 <= 2*Buy['RICE'] + 4.7*Buy['MILK'] + 15.3*Buy['BEEF'] + 17.8*Buy['PEPPER'] + 20*Buy['PEANUTS'] + 9*Buy['CHEESE'] + 20*Buy['NOODLES'] <= 3000;

Subject to Diet ['CAR.HY']:

72*Buy['CHEESE'] <= 9*Buy['BREAD'] + 10*Buy['RICE'] + 57*Buy['LETTUCE'] + 87*Buy['CHEESE'] + 343*Buy['GINGER'] + 65*Buy['PEANUTS'] + 57*Buy['LETTUCE'] + 87*Buy['CHEESE'] + 102*Buy['LIMES'] + 120*Buy['NOODLES'] <= 7600;

Subject to Diet ['RICE']:

3*Buy['BEEF'] + 3.5*Buy['WATERMELON'] + 1.69*Buy['LETTUCE'] + 1.39*Buy['NOODLES'] <= 3000;

Subject to Diet ['CHOLES']:

72*Buy['CHEESE'] <= 9*Buy['BREAD'] + 10*Buy['RICE'] + 57*Buy['LETTUCE'] + 350 <= 2*Buy['RICE'] + 4.7*Buy['MILK'] + 15.3*Buy['BEEF'] + 17.8*Buy['PEPPER'] + 20*Buy['PEANUTS'] + 9*Buy['CHEESE'] + 20*Buy['NOODLES'] <= 3000;

Subject to Diet ['CAR.HY']:

72*Buy['CHEESE'] <= 9*Buy['BREAD'] + 10*Buy['RICE'] + 57*Buy['LETTUCE'] + 87*Buy['CHEESE'] + 343*Buy['GINGER'] + 65*Buy['PEANUTS'] + 57*Buy['LETTUCE'] + 87*Buy['CHEESE'] + 102*Buy['LIMES'] + 120*Buy['NOODLES'] <= 7600;

Subject to Diet ['K']:

400 <= 10*Buy['RICE'] + 52*Buy['MILK'] + 80*Buy['COFFEE'] + 42*Buy['CALIFLOWER'] + 58*Buy['ORANGE'] + 31*Buy['BREAD'] + 0*Buy['OIL'] + 53*Buy['MUSHROOMS'] + 57*Buy['BEEF'] + 110*Buy['CHILI'] + 58*Buy['FISH'] + 98*Buy['TURKEY'] + 89*Buy['CREACKERS'] + 241*Buy['PEPPER'] <= 50000;
Subject to Diet ['A']:

\[ 350 \leq 0.7 \cdot \text{Buy['LIMES']} + 8.4 \cdot \text{Buy['NOODLES']} \leq 4000; \]
\[ 14 \cdot \text{Buy['COFFEE']} + 15 \cdot \text{Buy['CALIFLOWER')} + 3 \cdot \text{Buy['ORANGE')} + 80 \cdot \text{Buy['ICE')} + 92 \cdot \text{Buy['BREAD']} + 0 \cdot \text{Buy['OIL')} + 53 \cdot \text{Buy['EGG']} + 13 \cdot \text{Buy['MUSHROOMS')} + 51 \cdot \text{Buy['CHICKEN')} + 221 \cdot \text{Buy['BEEF']} + 99 \cdot \text{Buy['WATERMELON')} + 101 \cdot \text{Buy['CHILI')} + 18 \cdot \text{Buy['PUMPKIN')} + 97 \cdot \text{Buy['FISH')} + 1 \cdot \text{Buy['LITCHIS')} + 70 \cdot \text{Buy['TURKEY')} + 205 \cdot \text{Buy['NOODLES')} \leq 1200; \]
\[ 64 \cdot \text{Tanzila Yeasmin Nilu} \]

Subject to Diet ['C']:

\[ 350 \leq 0.7 \cdot \text{Buy['LIMES')} + 8.4 \cdot \text{Buy['NOODLES')} \leq 4000; \]
\[ 14 \cdot \text{Buy['COFFEE')} + 15 \cdot \text{Buy['CALIFLOWER')} + 3 \cdot \text{Buy['ORANGE')} + 80 \cdot \text{Buy['ICE')} + 92 \cdot \text{Buy['BREAD')} + 0 \cdot \text{Buy['OIL')} + 53 \cdot \text{Buy['EGG')} + 13 \cdot \text{Buy['MUSHROOMS')} + 51 \cdot \text{Buy['CHICKEN')} + 221 \cdot \text{Buy['BEEF')} + 99 \cdot \text{Buy['WATERMELON')} + 101 \cdot \text{Buy['CHILI')} + 18 \cdot \text{Buy['PUMPKIN')} + 97 \cdot \text{Buy['FISH')} + 1 \cdot \text{Buy['LITCHIS')} + 70 \cdot \text{Buy['TURKEY')} + 205 \cdot \text{Buy['NOODLES')} \leq 1200; \]
\[ 64 \cdot \text{Tanzila Yeasmin Nilu} \]

Subject to Diet ['NA ']:

\[ 10 \leq 0 \cdot \text{Buy['RICE']} + 0 \cdot \text{Buy['MILK')} + 0 \cdot \text{Buy['CHEESE')} + 0 \cdot \text{Buy['PEANUTS')} + 0 \cdot \text{Buy['PEPPER')} + 0 \cdot \text{Buy['BEEF')} + 0 \cdot \text{Buy['WA TERMELON')} + 0 \cdot \text{Buy['MUSHROOMS')} + 0 \cdot \text{Buy['CHICKEN')} + 0 \cdot \text{Buy['ICE')} + 0 \cdot \text{Buy['COFFEE')} + 0 \cdot \text{Buy['LIMES')} + 0 \cdot \text{Buy['NOODLES')} \leq 3000; \]

Subject to Diet ['B6 ']:

\[ 0 \leq 0 \cdot \text{Buy['RICE')} + 0 \cdot \text{Buy['MILK')} + 0 \cdot \text{Buy['CHEESE')} + 0 \cdot \text{Buy['PEANUTS')} + 0 \cdot \text{Buy['PEPPER')} + 0 \cdot \text{Buy['BEEF')} + 0 \cdot \text{Buy['WA TERMELON')} + 0 \cdot \text{Buy['MUSHROOMS')} + 0 \cdot \text{Buy['CHICKEN')} + 0 \cdot \text{Buy['ICE')} + 0 \cdot \text{Buy['COFFEE')} + 0 \cdot \text{Buy['LIMES')} + 0 \cdot \text{Buy['NOODLES')} \leq 1500; \]

Subject to Diet ['B12 ']:

\[ 75 \leq 0 \cdot \text{Buy['RICE')} + 0 \cdot \text{Buy['MILK')} + 0 \cdot \text{Buy['CHEESE')} + 0 \cdot \text{Buy['PEANUTS')} + 0 \cdot \text{Buy['PEPPER')} + 0 \cdot \text{Buy['BEEF')} + 0 \cdot \text{Buy['WA TERMELON')} + 0 \cdot \text{Buy['MUSHROOMS')} + 0 \cdot \text{Buy['CHICKEN')} + 0 \cdot \text{Buy['ICE')} + 0 \cdot \text{Buy['COFFEE')} + 0 \cdot \text{Buy['LIMES')} + 0 \cdot \text{Buy['NOODLES')} \leq 3500; \]

Subject to Diet ['Z']: 

\[ 100 \leq 0 \cdot \text{Buy['RICE')} + 0 \cdot \text{Buy['MILK')} + 0 \cdot \text{Buy['CHEESE')} + 0 \cdot \text{Buy['PEANUTS')} + 0 \cdot \text{Buy['PEPPER')} + 0 \cdot \text{Buy['BEEF')} + 0 \cdot \text{Buy['WA TERMELON')} + 0 \cdot \text{Buy['MUSHROOMS')} + 0 \cdot \text{Buy['CHICKEN')} + 0 \cdot \text{Buy['ICE')} + 0 \cdot \text{Buy['COFFEE')} + 0 \cdot \text{Buy['LIMES')} + 0 \cdot \text{Buy['NOODLES')} \leq 5000; \]

Subject to Diet ['ZN ']:

\[ 100 \leq 0 \cdot \text{Buy['RICE')} + 0 \cdot \text{Buy['MILK')} + 0 \cdot \text{Buy['CHEESE')} + 0 \cdot \text{Buy['PEANUTS')} + 0 \cdot \text{Buy['PEPPER')} + 0 \cdot \text{Buy['BEEF')} + 0 \cdot \text{Buy['WA TERMELON')} + 0 \cdot \text{Buy['MUSHROOMS')} + 0 \cdot \text{Buy['CHICKEN')} + 0 \cdot \text{Buy['ICE')} + 0 \cdot \text{Buy['COFFEE')} + 0 \cdot \text{Buy['LIMES')} + 0 \cdot \text{Buy['NOODLES')} \leq 5000; \]
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
    - BREAD 2
    - CAULIFLOWER 4
    - CHEESE 3
    - CHICKEN 3.5
    - CHILI 2
    - COFFEE 3
    - CREACKERS 3
    - EGG 2
    - FISH 3
    - GINGER 4
    - GRAVELEAVES 3
    - ICE 4
    - LETTUCE 3
    - LIMEJUICE 3
    - MILK 1
    - MUSHROOMS 3
    - NOODLES 2
    - OIL 2
    - ORANGE 3
    - PEANUTS 4
    - PEPPER 3
    - PUMPKIN 4
    - RICE 1
    - TOMATOES 2
    - WATERMELON 5

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


