**The Diet Problem**

Beichen Liang Mahir Bathija

University of Washington University of Washington

[liangbeichen@hotmail.com](mailto:liangbeichen@hotmail.com) [mahirbathija@gmail.com](mailto:mahirbathija@gmail.com)

August 2018

**INDEX**

1. Abstract (Mahir)………………………………………………………………………2
2. Introduction (Beichen)….…………………………………………………………….2
3. Background on Diet Problem (Beichen)……………………………………………..2-4
4. Assumptions (Mahir)…………………………………………………………………5
5. Mathematical Model (Beichen & Mahir)…………………………………………….6
6. Solution & Testing (Mahir)…………………………………………………………..7-8
7. Results (Beichen)……………………………………………………………………..9-10
8. Improvements (Beichen)……………………………………………………………..10-12
9. Conclusion (Beichen)………………………………………………………………...12-13
10. References (Beichen & Mahir)...…………………………………………………….13
11. Miscellaneous

* Data Entry and Collection of Food items (Beichen & Mahir)
* RStudio Code and Result Visualization (Mahir)

**Abstract**

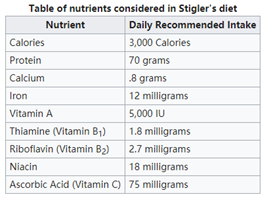
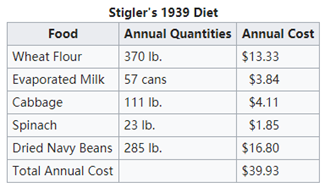
Given a set of foods, along with the nutrient information for each food and the cost per serving of each food, the objective of the diet problem is to select the number of servings of each food to purchase (and consume) so as to minimize the cost of the food while meeting the specified nutritional requirements. The importance of having a good diet is underestimated by many and can have heavy negative impacts on people's lives if not taken seriously. The problem is formulated as a linear program and the data we used was a manually written excel table of 171 different foods with nutritional information gathered from various grocery stores and the internet. We tested the results by trying the diet ourselves (and with some friends) as well by interviewing random students about their opinions on the diet. The solutions we found were successful in meeting dietary restrictions but were less than palatable and very dry.

**Introduction**

In the United States and worldwide, diet is a huge problem for everyone with the obesity rate rising rapidly. For people with no or lower income, like students, it is paramount to make a healthy diet at a reasonable price. While organic and healthy foods are sold at higher end grocery stores like Whole Foods with higher price, people with lower income are forced to consume fast and processed food within their price range. Meanwhile, there are many unhealthy foods with appealing taste, most of the fast food qualify. So, people with lower income are tempted to consume them, causing an unhealthy physical condition. Our goal is to construct a healthy and appealing diet at the lowest price possible.

**Background on Diet Problem**

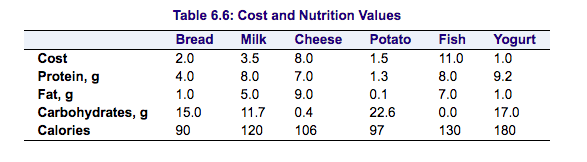
The diet problem was one of the first optimization problems studied in the 1930s and 1940s and it was motivated by the Army's desire to minimize the cost of feeding soldiers of the United States Army and airmen of the United States Army Air Forces in the field while still providing a healthy diet. It also has another name Stigler Diet, where it is named after George Stigler, who posed the following problem: *For a moderately active man weighing 154 pounds, how much of each of 77 foods should be eaten on a daily basis so that the man’s intake of nine nutrients will be at least equal to the recommended dietary allowances (RDAs) suggested by the National Research Council in 1943, with the cost of the diet being minimal?* The result was an annual budget allocated to foods such as evaporated milk, cabbage, dried navy beans, and beef liver at a cost of approximately $0.11 a day in 1939 U.S. dollars. Due to the lack of sophisticated method, Stigler used heuristic methods and get the solutions below.



Although, the diet is not applicable to life by its lack of variability, his work is considered to be some of the earliest work of linear programming. Later in the fall of 1947, Jack Laderman of the Mathematical Tables Project of the National Bureau of Standards used the simplex method newly developed by George Dentzig to solve Stigler's model. As the first "large scale" computation in optimization, the linear program consisted of nine equations in 77 unknowns. It took nine clerks using hand-operated desk calculators 120 man-days to solve for the optimal solution of $39.69. As the growth of World Wide Web in the late 20th century, more interactive case studies were posted. Joseph Czyzyk and Timothy J. Wisniewski from Mathematics and Computer Science Division Argonne National Laboratory derived an interactive calculator with the help of World Wide Web to deliver better educational experience to students.

Given a set of foods with the nutritional information for each food and the cost per serving of each food, the objective of the diet problem is to select the number of servings of each food to purchase (and consume) to minimize the cost of the food while meeting the specified nutritional requirements.

A small example of our problem can be considered below. Give, in the table are 6 different food with their cost and four nutrition values.



The objective is to find a minimum-cost diet that contains at least 300 calories, not more than 10 grams of protein, not less than 10 grams of carbohydrates, and not less than 8 grams of fat. In addition, the diet should contain at least 0.5 unit of fish and no more than 1 unit of milk.

**Let the number of units of food to be consumed be:**

Bread = x1

Milk = x2

Cheese = x3

Potato = x4

Fish = x5

Yogurt = x6

**Objective function to minimize:**

2x1 + 3.5x2 + 8x3 + 1.5x4 + 11x5 + 1x6

**Subject to:**

90x1 + 120x2 + 106x3 + 97x4 + 130x5 + 180x6 >= 300 (calorie constraint)

4x1 + 8x2 + 7x3 + 1.3x4 + 8x5 + 9.2x6 <= 10 (protein constraint)

15x1 + 11.7x2 + 0.4x3 + 22.6x4 + 17x6 >= 10 (carbohydrate constraint)

1x1 + 5x2 + 9x3 + 0.1x4 + 7x5 + 1x6 >= 8 (fat constraint)

x5 >= 0.5 (fish constraint)

x2 <= 1 (milk constraint)

**Throwing these constraints into LPSolve, we get the following results:**

Bread = x1 = 0

Milk = x2 = 0.449499

Cheese = x3 = 0.5

Potato = x4 = 0.053599

Fish = x5 = 1.865168

Yogurt = x6 = 0

In order to find a low price yet still healthy diet, we want to compare the prices as well as the nutritional values. Using recommended daily nutrition intake from the National Institute of Health under Department of Health & Human Services, we put all the data we have together to form a linear program that meets the recommended values while still keeping the price down.

The next section will describe the assumptions we made in the study. Section Mathematical model formulate diet problem as a linear program and topics on implementation details on the mathematical model. The following section poses the initial basic solution and testing conducted on that. The results and potential improvements are then illustrated. The ending section makes concluding remarks about the Diet problem and the menu.

**Assumptions**

To begin with, the diet problem is a very complex one because there are tens of thousands of types of different types of ingredients, meals and foods. Obviously, it would be impractical for us for collect data on every single one of these, so we made a list of 100 different ingredients and foods that are commonly consumed across the United States.

Another component of the diet problem is figuring how many of each types of nutrient does a single person need in a given day. Once again, it would be way too difficult for us to gather and integrate information about every single one of these little constraints, so we included the top 14 most important ones (protein, carbohydrates,

fat, cholesterol just to name a few).

Our next step was to find out how much of each of these nutrients does one need in a given day. We came across a wide variety of such requirements for different genders, ages and conditions (such as male, female, 19-30, infant, pregnant etc.). To generalize and simplify our problem further we assumed the audience of this article to be a male between the age of 19-30 years and hence the upper and lower bounds for this nutrient are given based on a 2,000 calorie (the most common calorie level diet eaten by this category of people) eaten by such individuals.

Our next assumption was to assume that the minimum serving of a certain food one would eat is one. If we had not assumed that, then the linear program software we used would be able to spit out numbers like 0.290 and 0.001. While it is possible for people to eat a half/third portion of foods, it is unlikely for anyone to eat portions of food such as the decimal numbers mentioned above. But since our model is allowed to spit out real numbers, we could also get numbers like 1.192 which is basically not possible or convenient to measure. This could be a limitation in our model that we need to re touch up on after our testing phase.

After running tests with these assumptions, we came to a conclusion that if we need a more accurate answer we need to do the following:

* Include a more comprehensive list of foods (somewhere in the tens of thousands) so that different (more) foods can be combined in order to make a variety of meals.
* Include possibly every single nutrient that is required by the human body that we did not include. This could be things like Vitamin A, Vitamin B12 etc.
* Try and break down some of the bigger nutrients into smaller components. One such example in our model is the breaking down of fat into monounsaturated fat, polyunsaturated fat and trans-fat.
* Include a sort of interactive feature that allows the user to alter the age range, gender, calorie level diet and current body status in order to get a more personalized answer.
* Include a way for the model to spit out numbers that are more suited to convenience like 1.5, 2.5 etc.

**Mathematical Model**

The main objective of our mathematical model to minimize the total cost that an individual spends on food in a single day. As stated in the section above, we further simplified our model on the testing ground to see what method we should implement to best solve the diet problem. By taking just a handful of different foods and a couple of nutrients as constraints, we came to the conclusion that a basic form of Linear Programming would be best suited to solving our problem.

The Diet Problem can be formulated mathematically as a linear programming problem:

Sets:

Parameters:

* ,

Variables:

Objective Function: Minimize the total cost of the food

Minimize

Constraint Set 1: For each nutrient , do not exceed the maximum allowable amount

Constraint Set 2: For each nutrient , meet the minimum required level

Constraint Set 3: For each food , do not exceed the maximum allowable amount

Constraint Set 4: For each food , include the minimum servings of the food

**Solution of Mathematical Problem & Testing**

As stated previously in our paper, we used basic concepts like Linear Programming to solve the diet problem. It seemed like converting it into a Mixed Integer Linear Program (MILP) would be the right thing to do and so we did. But during the testing phase, we came across several difficulties which we had to tackle head on.

To begin with, we had about 22 nutritional constraints, which translates to 44 hard constraints (upper and lower bound of each nutrient). During our brainstorming phase, we did not foresee the fact that so many constraints could possibly break our Linear Program due to the nature of the model (only having 100 items at first and the model being only be able to spit out integer values). It was not unlikely that adding so many constraints to our program would make it unsolvable, which it did. Therefore, we decided to reduce the number of hard constraints down to 10 nutrients (20 hard constraints upper and lower bounds of each nutrient) and make the strain on the linear program slightly less.

Finally, this move made us gain some progress and momentum and was the push we needed to go in the right direction. Although our linear program was taking longer than we expected to execute we hoped to add more constraints and give our users the luxury of being able to eat multiples of quarter portions to make our solution more akin to the real world. At first we believed that adding more food items was a way in which we could add more constraints because our intuition was that the program was breaking due to less variety of foods. So we added 71 more food items to our database and tested our program again. Although we were able to add to more nutrients (4 hard constraints upper and lower bound of each of the 2 nutrients) this still did not lower the runtime, give our users the capability to eat quarter portions or allows us to add more nutritional constraints without breaking the program.

In order to fix this and the fact that we wanted to add more constraints while make the program runtime as efficient as possible we decided to change our techniques that we were using from MILP to use real numbers instead. Below I have listed the advantages of using real number instead of integers in our linear program:

* Our program runs exponentially faster. Our previous runtime with 22 constraints was about ~15 seconds which is fairly bad. After switching from integers to real numbers and adding 2 more constraints, our program basically takes negligible time to run ~0.1 seconds (almost too small to estimate without inbuilt functions).
* We are now able to offer servings in multiples of quarter fractions to our users which happens often in the real world.
* Because of the advantage stated above, our results are more likely to include a more diverse variety of foods to consume in a day.
* We are to able to add 7 more nutrient constraints (making it a total of 36 hard constraints upper and lower bounds of each nutrient).
* The only disadvantage of using real numbers is the fact that our program sometimes gives us inconvenient numbers such as 0.123, 0.0001, 0.872 which are not possible to consume in the real world. A simple fix for this was to round the answer to the closest quarter fraction (0.25, 0.5, 0.75, 1, ...). Although this will not give us the exact answer we are looking for, it is a fairly close approximation to the real answer. The cost difference is about 10% higher or lower compared to the original answer which considering the low numbers isn’t too bad.

Hence, we were able to improve our model significantly by adding more nutrients as constraints and different serving sizes which ultimately lead to more variety in the food choices. Furthermore, we did all programming, data manipulation and linear programming in R studio with the help of the lpSolveAPI package to solve our linear program. The following is the pseudocode for how we did the tasks mentioned above:

Read in csv file into variable - dat

For each nutrient i: i = dat[nutrient i]

Set lower and upper bounds for serving sizes

Make lp model

Add upper and lower constraint for each nutrient i

Set row and column names

solve (lp model)

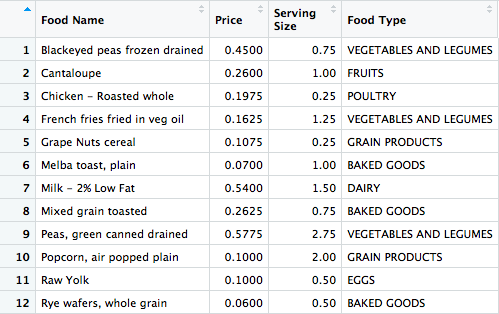
Round results to closest quarter multiple

Write lp to external text file and save to computer

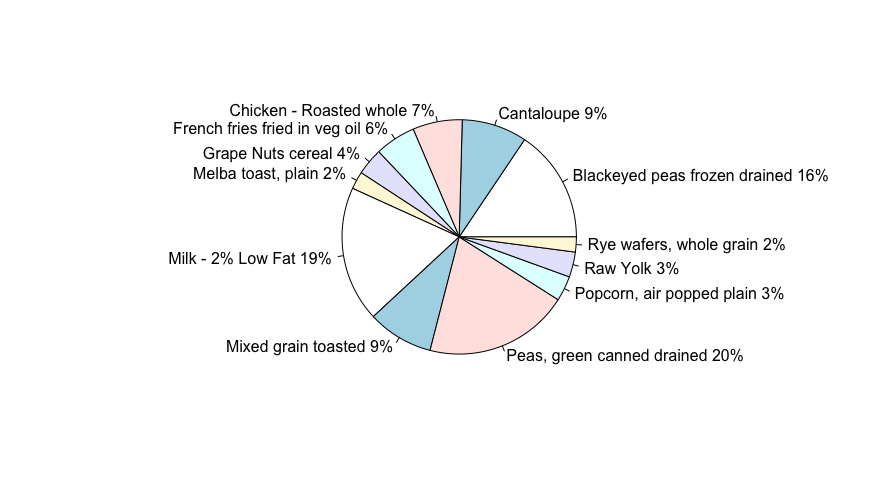
(A more detailed version of the code with comments can find found in the RStudio file attached)

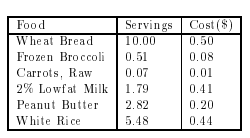
**Results**

The total cost is $2.89 per day with 171 foods and two assumptions. First, the maximum allowable servings of every food is five. Second, the servings are rounded to quarter servings.

****

**Cost Breakdown of the diet**

****

Compared to what Stigler originally proposed, our menu is significantly more diverse. In our diet, there are twelve different food while Stigler’s menu only has five. Later in the interactive study, the authors used newer data to get a different solution. 

However, since we restricted servings with a smaller upper bound, we also attain more varieties in our diet. The group, along with two other friends tested the diet on 11th Aug 2018. Although, the thing looks good, after the day, we reached a unanimous consensus that a day with no beverage cannot be tolerated. Meanwhile, Mahir cannot live a day without apple pie. Thus, even if he knows that the diet is healthy and costs the least, he has bare interest in keeping up with it.

**Improvements**

Although our results gave us more food items than Stigler’s our menu still lacked variety. There are three veggies while no beverage. Taking food across different categories has larger probability to be appealing to people. During the testing phase, we added 71 more ingredients, however, that only changed the variety of our diet trivially. Thus, encoding the variety requirement as a constraint is an obvious improvement. We decided that the 6 most common varieties on should have in a day in order to make the diet palatable, were fruits, meat, beverages, fats and oils, dairy and vegetables and legumes.  
  
To encode this requirements, new parameters are added:

New sets:

New Parameters:

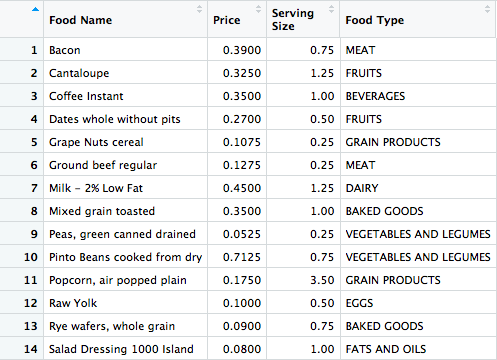
New Constraints:

Constraint Set 5: For each food category , do not exceed the maximum variety

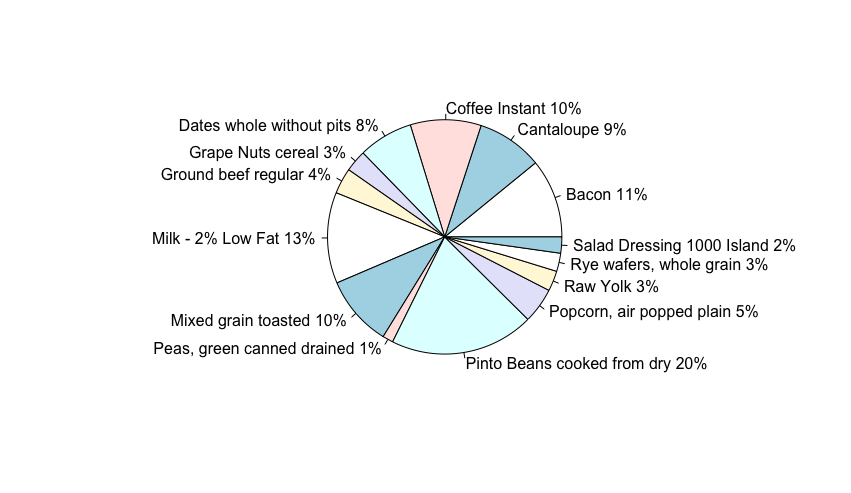
Constraint Set 6: For each food category , include at least the minimum number of foods from the category

By setting initial minimum amount to be one, the program guarantees that in the final diet, there exists at least one servings of food from each category. For example, when we set minimum required serving of major food categories, meat, fruits, beverages, dairy, fats and oils and vegetables and legumes to be one, we got the following result:

The total cost is $3.58 per day with 171 foods and two assumptions. First, the maximum allowable servings of every food are five. Second, the servings are rounded to quarter servings. As we can see, we got instant coffee as an additional beverage to our diet.



**Cost Breakdown of the diet**



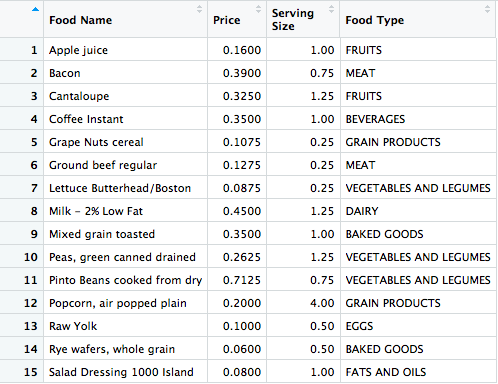
Another thing to note here is the personalization of the diet. Since there is an optimal solution under the fixed linear program, no matter what kind of food the people like, the program will always give the same diet solution. To solve this problem, users would be prompted to take a survey asking their food preference. In the survey, each food will be rated on a 0-10 scale about how much you like about it, and those levels will be converted into a new parameter that exerts an effect on the linear program and the optimization.

New Objective Function:

Minimize

New Parameter:

Using the rating of food, the objective function hedges the personal preference on diet. If the user likes something a lot, he can simply rate that to be level 10 and the preference can be easily reflected. After we figure out, what is the exact servings of each food, we can easily track back the minimum cost by reverse calculation. For example, say someone loves apple juice. So, suppose we rate apple juice 10, after conducting optimization, the new menu now contains a serving of apple juice. The new diet plan based on all original assumptions cost $3.86 ($3.76 – 0.16(adjusted value of apple juice after rating) + 1\*0.26 (original price of apple juice)) per day.



**Conclusion**

For the first part of our problem, we have shown that linear programming can be used to effectively implement a diet plan that meets the recommended daily intake of a person. We were able to input all the data about the caloric intake, price, and nutritional value, and find a nutritional meal at the lowest possible price. Although it is a very strange meal, but it will satisfy the daily recommended intake of all the nutrition groups. In the improvement section of the problem we aimed to find more variety and more personalized meal options. With budget, nutritional and the additional variety and personalization constraints, we were able to build up a diet with more variety and personalization to optimize the cost. As we can see, adding variety and personalization can slightly increase the cost. In conclusion, the most basic meal will satisfy the recommended daily intake for a low price, but if you crave more variety in your meals there will be an extra cost. Our model can be generalized to apply to build diet plan for general public and those groups with special needs, including military, vegetarians, etc.

**References**

“The Diet Problem.” *NEOS*, neos-guide.org/content/diet-problem.

Czyzyk, Joseph. *Diet*. ftp.mcs.anl.gov.

The Diet Problem

Interfaces, Vol. 20, No. 4, The Practice of Mathematical Programming (Jul. - Aug. 1990), pp. 43-47

George B. Dantzig

Diet Optimization Problem

<http://users.iems.northwestern.edu/~armbruster/2009iems310/project/diet.pdf>

Francesca Ferrero, Emilie Hsieh, Ariel Wagner

The Diet Problem: A WWW-based Interactive Case Study in Linear Programming (http://www.mcs.anl.gov/home/otc/Guide/CaseStudies/diet/)

Joseph Czyzyk and Timothy J. Wisniewski