

The Diet Problem in Developing Countries

Matthew Grattan^{*}

George Ho[†]

ChE488: Convex Optimization Techniques

The Cooper Union for the Advancement of Science and Art,

41 Cooper Square, New York, NY 10003, US

November 13, 2018

^{*}grattan@cooper.edu

[†]ho@cooper.edu

Contents

Abstract	2
1 Introduction	1
2 Problem Description	1
3 Results	3
3.1 Solver	3
3.2 Optimal Solution	3
3.3 Sensitivity Analysis	4
4 Conclusion	4
4.1 Discussion	4
4.2 Future Directions	5
5 Appendix	6

Abstract

The concentration of an unknown FeCl_3 solution was determined to be 60 ± 2 mM using linear sweep voltammetry. Cyclic voltammetry was used to determine an experimental reduction potential of 0.813 V for the $\text{Fe}^{3+}/\text{Fe}^{2+}$ half-reaction, which had an error of 5.4% from the literature value of 0.771 V.

1 Introduction

The diet problem is a classic linear programming problem (LP) whose solution is cheapest diet that satisfies a set of nutritional constraints. The problem was first posed by Jerry Cornfield for the US Army during World War II¹ to find “a low cost diet that would meet the nutritional needs of a GI soldier.”² George Stigler, an economist, offered a heuristic solution to the diet problem for 77 foods in 1945, but it was not necessarily the minimum cost possible.^{2,3} In 1947, Jack Laderman solved Stigler’s diet problem with George Dantzig’s simplex method,² one of the first algorithms to solve LPs efficiently.⁴ LPs, including the diet problem, were important for military logistics, and their solutions helped minimize costs associated with the allocation of supplies. Many solution methods were developed during World War II.⁴

The diet problem is interesting to anyone trying to find the cheapest diet given a set of foods and nutrients. The problem is of interest because the constraints are quite flexible; they can be as simple as calories or macronutrients, but can also include vitamins and minerals, total food weight, and variety of foods in the solution set. Contemporary approaches to the diet problem account for specific health concerns like glycemic load for people with diabetes,⁵ or seek to minimize environmental impact by accounting for carbon emissions, nitrogen release, water usage, and land usage.⁶ Other novel approaches employ fuzzy logic to model uncertainty, like fluctuations in pricing.⁵

In this paper, we are particularly interested in solving the diet problem for different developing countries. Food supplies vary based on region, so the regional availability of certain foods further constrains the problem. We would like to minimize the cost of subsistence in 1457 markets across 74 developing countries. We are constraining the problem with only four nutrients (carbohydrates, fat, protein, and fiber) and a set of 85 unique foods. Not all of the foods are available in each market.

2 Problem Description

We acquired dataset on world food prices compiled by the World Food Program from Kaggle.⁷ The data set contained the prices of 321 commodities from 1457 markets across 74 developing countries. We removed the commodities that were not foods and consolidated subtypes of foods into one entry (i.e. we treated “Bread (wheat)” and “Bread (rye)” as simply “Bread”). This data processing resulted in a final list of 85 foods.

We acquired nutrition data on carbohydrates, fat, protein, and fiber for the reduced set of 85 foods from the USDA Branded Food Products Database.⁸ We chose not to include vitamins and minerals to focus what would likely be staple foods—and for convenience as the data was entered by hand.

In general, LPs can be solved using a variety of methods, depending on the size of the problem. Brute-

force is conceptually the simplest but also the most time consuming: A basic solution is computed for each of the n vertices of the feasible set, resulting $n!$ linear systems.⁴ A popular solution method for LPs is the simplex algorithm, which finds the minimum value by calculating the objective starting at one vertex and moves in the direction of the adjacent vertex with the lowest objective value. Interior-point methods were developed more recently in the 1980s and are effective for very large LPs.⁴

Our problem is actually a set of 1457 LPs, each with 85 variables and four constraints. We chose the simplex algorithm as our solution method and implemented it using the built-in LP solver from the SciPy package for Python (`scipy.optimize.linprog`).⁹

We can pose the problem in the following way:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) = c^\top x \\ & \text{subject to} && Ax \leq b \\ & && x \geq 0 \end{aligned}$$

where $c \in \mathbb{R}^N$, $b \in \mathbb{R}^M$, and $A \in \mathbb{R}^{M \times N}$. For our problem specifically, $N = 85$ and $M = 4$. The vector c represents the cost of each of the 85 foods, the vector b represents the recommended daily intake of each nutrient, and the matrix A represents the amount of each nutrient in each food. The decision variable $x \in \mathbb{R}^N$ represents the amount of food purchased in kilograms.

Let $f(x)$ be the cost function (literally the cost of food) we seek to minimize, and define the feasible set as

$$\Omega = \{x \in \mathbb{R}^N : Ax \leq b, x \geq 0\}$$

To prove the existence of a feasible solution x^* to our problem, we first apply the Weierstrass Extreme Value Theorem.⁴

The feasible set Ω is nonempty because it contains at least the zero vector. Suppose $x = 0$, then

$$Ax = 0 \leq B$$

All inequalities are nonstrict, so Ω is a closed polytope in \mathbb{R}^N . The set Ω is also bounded. Let

$$\epsilon = 1 + \max_i |b_i|$$

and define the open ball of radius epsilon centered at the origin as

$$B_\epsilon(0) = \{x \in \mathbb{R}^N : \|x\| < \epsilon\}$$

then $\forall x \in \Omega, x \in B_\epsilon(0)$ also. This implies that $\Omega \subseteq B_\epsilon(0)$. Therefore, Ω is compact, as it is closed and bounded.

The objective $f(x) = c^\top x$ is a linear functional, that maps vectors in \mathbb{R}^N to scalar values in \mathbb{R} . $\Omega \subset \mathbb{R}^N$, so f is defined on Ω . Matrix multiplication is continuous; thus, f is continuous.

The conditions of the Weierstrass theorem are satisfied, so there exists $x^* \in \Omega$ that minimizes $f(x)$.

We next show that x^* lies on the boundary of Ω by applying the first-order necessary conditions for interior points. $f \in \mathcal{C}^1$ because f is simply matrix multiplication. $\nabla f(x) = c \forall x$, but for interior points that satisfy the first order necessary conditions, $\nabla f(x^*) = 0$. Therefore, $x^* \in \partial\Omega$.

3 Results

3.1 Solver

LPs are extremely well-studied and well-understood, with open-source implementations of solvers available. We use the SciPy implementation of the simplex algorithm (`scipy.optimize.linprog`) to solve our optimization problems.⁹

The problems were fairly small. We considered 85 possible foods, and 4 possible nutrients: this means that each problem had 85 variables and 4 constraints. Note that even though not all markets supply all foods, we set up the problem as if they did, but with a unit price of 999999. Thus, each problem has exactly 85 variables and 4 constraints. The bulk of the computation time comes down to the number of problems we solved: there are 1457 unique markets in developing countries for which we have data. In total, solving all optimization problems took less than 2 minutes.

Our code and data can be found on GitHub.¹⁰

3.2 Optimal Solution

Since there are 1457 different optimization problems (each with a minimum cost as expressed in the local currency, and not normalized to one single currency), it is infeasible to go over each optimal solution. Nevertheless, we make a few remarks. No optimization problem was infeasible, nor were any problems unbounded: the optimization algorithm terminated successfully for all problems. Furthermore, the solutions tended to be very sparse: that is, most foods would not be bought at all, and all nutrients would be supplied by buying some quantity of one or two foods.

3.3 Sensitivity Analysis

It is important to gain some sense of how sensitive the solution is to changes in the parameters. However, due to the sheer number of optimization problems, and the large number of variables within each one, it is very difficult to perform a sensitivity analysis of each problem individually by perturbing the price/nutrition information. Thus, for the purposes of this report, we limit ourselves to only one optimization problem (that is to say, only one market in one locality in one country), and we only perturb the prices and nutrition values of the foods demanded by the efficient diet. In other words, if a food is not in the efficient diet, we do not bother perturbing its values. Finally, we only perturb parameters by either 10% or 50%. Sensitivity analysis of the solution is an intense effort to begin with, so we make these simplifying assumptions in the interest of brevity.

One result of this sensitivity analysis was that decreasing the nutritional value of pasta by either 10% or 50% led to lentils being included in the efficient diet, instead of pasta. From this, we can infer that lentils are nutritionally a suitable substitute for pasta, if only they were slightly more nutritious. The exact same phenomenon occurs if we increase the price of pasta: again, we see that lentils are “like” pasta, but just slightly more expensive. Besides this observation, all other solutions to the perturbed optimization problems involve only eggs and pasta.

Another interesting observation is how important pasta is to this particular efficient diet: any changes to pasta’s price or nutritional value impacts the cost of the resulting diet much more than an equivalent change to the price or nutritional value of eggs. For example, halving the price of eggs merely reduces the cost of the diet from 277 to 251 Armenian dram: barely a 10% decrease. However, halving the price of pasta reduces the cost of the diet from 277 to 165 dram: just over a 40% reduction.

4 Conclusion

4.1 Discussion

As explained above, it is infeasible to discuss the numerical solutions to each of the 1457 problems. Nevertheless, we will discuss one particular solution, as its interpretation within the context of the original problem is the same across all such solutions. Here, we consider the efficient diet for the market in Armavir, in the locality of Lori, in Armenia.

This solution entails 0.66 for pasta, and 0.56 for eggs, with the objective function taking on the value of 278. The way one interprets this is that the efficient diet entails eating 0.66 kilograms of pasta, and 0.56 kilograms of eggs every day, with the cost totally 278 Armenian Dram each day (this is roughly 0.57 US

dollars).

Put in this light, it is painfully clear that such a diet is not sustainable: it is probably neither appealing nor ideal to eat the same food every day, and there may not even be a recipe to cook a proper meal with just those two ingredients. We discuss this, and many other shortcomings and failings of our project, below.

We also make several other remarks on the solutions: we find that the most demanded foods (assuming that everybody follows the efficient diet) are rice, cowpeas, maize and wheat, each of which have a demand in excess of 1000 kilograms every day. If domestic supply of these foods fall short of this figure, then the developing country will need import them, which is not ideal. Furthermore, more than half of all foods under consideration are not demanded at all, from any market.

4.2 Future Directions

There are several shortcomings of this project, and several other directions for future work. We discuss each one in turn.

Firstly, we used food price data from developing countries, but we use used nutrition data from the USDA. There is some incongruity there: we are effectively assuming that food in developing countries has the same nutritional content as their counterparts in developed countries. This is probably true (or true enough) for staple foods such as chicken and beef. However, foods such as rice (or any other crop) is known to have different nutritional content depending on the nutritional content of the soil in which it is grown. Thus, it would be best to obtain a data set that is representative of food in developing countries, instead of having to stitch together two separate data sets.

Secondly, our original data set leaves much to be desired: it only covers 85 foods. It would be ideal to obtain a more comprehensive data set of all foods available in developing countries.

Thirdly, we implicitly assume that it possible to buy fractional amounts of each food. This is clearly not true: one cannot buy anything but an integer number of chickens, and it is improbable that one can buy any quantity of rice. Thus, we should treat the diet problem as an integer linear problem (ILP), or, in the worst case, a mixed-integer linear problem (MILP), in order to account for the specific quantities in which one purchases food.

Fourthly, in the interest of simplicity, we simply discarded any additional information about the food supplied in parenthetical statements. Most such parenthetical statements disambiguated whether or not the particular food was imported, or the exact species of the food, and we thought it justifiable to discard this information. However, sometimes these parenthetical statements differentiated between the kind of poultry (e.g. chicken, turkey, duck etc.), or even different kinds of meats (e.g. beef, pork, lamb etc.). These

distinctions are very important to make, and is something we should take into account.

Finally, we only consider the four most important nutrients. However, having seen that all markets are capable of supplying a feasible diet, it is now more interesting to see if it is possible for people in developing countries to maintain a diet that is considered healthy and recommended in a developed country: in other words, include the recommended daily intake of all vitamins and nutrients into the optimization problem.

References

- [1] van Dooren, C. A Review of the Use of Linear Programming to Optimize Diets, Nutritiously, Economically and Environmentally. *Frontiers in Nutrition* **2018**, 5, 48.
- [2] Dantzig, G. B. The Diet Problem. *Interfaces* **1990**, 20, 43–47.
- [3] Stigler, G. J. The Cost of Subsistence. *Journal of Farm Economics* **1945**, 27, 303–314.
- [4] Chong, E. K.; Zak, S. H. An introduction to optimization.
- [5] Bas, E. A robust optimization approach to diet problem with overall glycemic load as objective function. *Applied Mathematical Modelling* **2014**, 38, 4926 – 4940.
- [6] Gephart, J. A.; Davis, K. F.; Emery, K. A.; Leach, A. M.; Galloway, J. N.; Pace, M. L. The environmental cost of subsistence: Optimizing diets to minimize footprints. *Science of The Total Environment* **2016**, 553, 120 – 127.
- [7] Boysen, J. World Food Prices. <https://www.kaggle.com/jboysen/global-food-prices>.
- [8] USDA Food Composition Databases. <https://ndb.nal.usda.gov/ndb/search/list>.
- [9] `scipy.optimize.linprog`. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html>.
- [10] Ho, G. diet-problem. <https://github.com/eigenfoo/diet-problem>.

5 Appendix

All our code can be found on the GitHub repository.¹⁰ In the interest of brevity we only attach the code of `all_countries.ipynb` here.

```

In [1]: import numpy as np
        import pandas as pd
        from scipy.optimize import linprog

In [2]: NUTRITION_FILE = '../data/nutrition.csv'
        PRICING_FILE = '../data/pricing.csv'

        # If a particular market does not have a food,
        # we set its price to be "infinity" (i.e. 999999).
        INFINITY = 999999

In [3]: nutrition = pd.read_csv(NUTRITION_FILE, index_col=0)
        pricing = pd.read_csv(PRICING_FILE, index_col=0)

In [4]: grouped = pricing.groupby(['country_name', 'locality_name', 'market_name'])

In [ ]: solns = {}

        for market, idx in grouped.groups.items():
            df = pricing.loc[idx]

            A_ub = -np.transpose(nutrition.values)
            b_ub = -np.array([65, 300, 25, 50])

            # Construct c appropriately (i.e. add 0s to the appropriate foods)
            c = pd.Series(data=INFINITY*np.ones(84), index=nutrition.index)
            c.loc[df.commodity_name] = df.price.values
            c = c.values

            solns[market] = linprog(c, A_ub, b_ub)

In [ ]: # Collect optimization status, minimum value and minimum into one array
        data = np.hstack([
            np.transpose(
                np.vstack([[soln.status for soln in solns.values()],
                           [soln.fun for soln in solns.values()]]))

```

```
    ),  
    [soln.x for soln in solns.values()]  
])
```

```
In [ ]: df = pd.DataFrame(data=data,  
                           index=solns.keys(),  
                           columns=['status', 'fun'] + nutrition.index.tolist())
```

```
In [ ]: df.to_csv('all_countries.csv')
```