

Assignment 1: The Diet Problem Revisited

Part 1:

For this assignment, I will be using 5 packaged items to implement The Diet Problem. The documentation for the five packaged food items has been included in the github repository. In addition to the labeling of the files, here are the items and their prices per serving:

- Devanco Beef Bacon - \$0.85/serving (\$8.49 / 10 servings)
- Chicken Breasts – \$0.46/serving (\$6.97 / ~15 servings)
- Kerrygold Unsalted Butter – \$0.37/serving (\$5.99 / 16 servings)
- O Organics Quinoa – \$0.70/serving (\$6.99 / 10 servings)
- Cocoa Puffs Cereal – \$0.85/serving (\$6.79 / 8 servings)

The prices per serving have been calculated using the price of the packaged item divided by the number of servings per package the nutritional facts mention. It is worth noting that the beef bacon and chicken breasts labels only mention serving sizes in weight, which I used to divide the package weight to find the number of servings per package.

Part 2:

Here is the linear programming problem in standard form:

Decision Variables:

- X_1 – Beef Bacon
- X_2 – Chicken Breasts
- X_3 – Butter
- X_4 – Quinoa
- X_5 – Cereal

Objective Function:

$$\text{Minimize } Z = 0.85x_1 + 0.46x_2 + 0.37x_3 + 0.7x_4 + 0.85x_5$$

Constraints :

These inequalities are for sodium, energy, protein, vitamin D, calcium, iron, potassium and lastly non-zero requirements, in that order:

$$270x_1 + 74x_2 + 0x_3 + 0x_4 + 130x_5 \leq 35000$$

$$45x_1 + 165x_2 + 100x_3 + 170x_4 + 140x_5 \geq 14000$$

$$5x_1 + 31x_2 + 0x_3 + 6x_4 + 2x_5 \geq 350$$

$$0x_1 + 0x_2 + 0x_3 + 0x_4 + 4x_5 \geq 140$$

$$2x_1 + 11x_2 + 0x_3 + 20x_4 + 130x_5 \geq 9100$$

$$0x_1 + 1x_2 + 0x_3 + 2.1x_4 + 3.6x_5 \geq 126$$

$$58x_1 + 0x_2 + 0x_3 + 250x_4 + 10x_5 \geq 32900$$

$$x_1, x_2, x_3, x_4, x_5 \geq 1$$

This linear programming problem is designed to calculate a weekly diet using at least one of each of the 5 designated packaged food items with the goal finding the cheapest diet that will satisfy the nutritional needs within a week. The Python PuLP code and the output of the code have been included in the github repository under “Assignment_1_PuLP.py” and “PuLP_Output.txt” respectively.

Part 3:

Running the PuLP code, we get the following results for the weekly number of servings per each food item and the weekly cost of all the food items:

Optimal servings for each food item for minimum cost:

Devanco_Beef_Bacon: 0.0 servings

Chicken_Breasts: 0.0 servings

Kerrygold_Butter: 0.0 servings

O_Organics_Quinoa: 129.59752 servings

Cocoa_Puffs_Cereal: 50.06192 servings

Total weekly cost: \$133.27

The results show that to get the nutritional needs with minimum cost, I could consume 0 servings of beef bacon, chicken breast and butter and 129.6 servings of quinoa and finally 50.1 servings of cereal. The total cost of all these servings would come to \$133.27 per week.

Part 4:

In this part, I have included an additional constraint (mentioned also as the last constraint in part 2) that requires each packaged item to be used at least once each week. The PuLP code in the github repository is of this revised version and the output of the code can be found in "PuLP_Output_Revised.txt" with the following output:

Optimal servings for each food item for minimum cost:

Devanco_Beef_Bacon: 1.0 servings

Chicken_Breasts: 1.0 servings

Kerrygold_Butter: 1.0 servings

O_Organics_Quinoa: 129.36811 servings

Cocoa_Puffs_Cereal: 49.997214 servings

Total weekly cost: \$134.74

The additional constraint does not seem to have a big impact on the results. The only differences are the 1 additional serving for each of the 3 items that had 0 serving in the original result as well as the small changes in the other two items and the total weekly cost of the items. In comparison to the original results, these additional constraints did not change the results while introducing minimal variety in the diet. To further introduce variety to the diet, we can add additional food items, increase the minimum serving for each item further, or implement more constraints such as fibers or fats.

Part 5:

I have selected ChatGPT as the LLM model (<https://chatgpt.com/>). For this part, I initially started the conversation with the agent by narrowing its focus by telling it to pretend to be a data scientist, and describe the problem to tackle. Once provided with an approach and code, I specified some of the requirements for our assignment approach, such as using PuLP or specifying the nutritional constraints. Once I refined the requirements, the agent was able to provide a code similar to what I wrote for the assignment. There is a chance that as the problem is a popular one, the training of the agent probably includes solution code, which is why it was able to provide answers that are ideal. I think the agent was able to successfully provide an adequate solution to The Diet Problem. When tackling problems such as this, it is important to define a focus for the agent, followed by a general direction with goals and tasks. I seem to find success when I allow the agent to provide an initial response, followed by prompting for modifications over the initial response as well as any additional specifications. This allows the agent to work on its own response to better tailor the answer to the needs of the task at hand. I should point out that I grouped up several prompts and ended up with 2 prompts to get a relatively desired answer. More prompts could result in a further performance increase with the agent.