Optimizing a Personalized Diet: A Linear Programming Approach to Nutritional Economics

Northwestern University, MSDS 460: Decision Analytics

Homework Assignment 1: Linear Programming --The Diet Problem

Janvier 14, 2024

1. Introduction

In this paper, I apply linear programming to optimize my weekly diet. My goal is to identify a cost-effective food combination that fulfills my nutritional needs. This requires selecting food items, calculating costs, and setting constraints in a linear programming framework.

2. Methods

Part 1: Selection and Documentation of Food Items To begin, I selected five staple items from my pantry: almonds, brown rice, Greek yogurt, pizza, and tofu vegetables & hash brown. I captured images of the nutrition labels to establish a database of their costs and nutritional content. Each item's cost per serving was meticulously calculated, forming a critical component of the linear programming model. Part 2: Formulating the Linear Programming Problem With the decision variables representing weekly servings, I crafted an objective function to minimize my total food expenditure. Simultaneously, I constructed constraints for the model based on nutritional guidelines to ensure a balanced diet. Part 3: Implementation Using AMPL Using AMPL, I translated my dietary constraints and objectives into a solvable linear programming problem. The model was solved with the aid of the PuLP solver, chosen for its robustness and efficiency in handling such optimization problems. Part 4: Solution Analysis The initial solution, as computed by AMPL, suggested an imbalanced diet consisting solely of pizza, based on the cost and nutrient content parameters. This underscored the limitations of a cost-centric approach. Part 5: Revising the Problem for Dietary Variety Recognizing the importance of dietary diversity, I revised my model to include a constraint that mandated at least one serving of each food item. This adjustment was crucial for achieving a more realistic and varied diet plan.

3. Results

Part 1: Cost Per Serving Calculations The calculated costs per serving were as follows: almonds at \$0.62, brown rice at \$0.74, Greek yogurt at \$1.49, pizza at \$2.87, and tofu vegetables & hash brown at \$6.99. These figures were foundational for the subsequent optimization process. Part 2: The Linear Programming Model I described the linear programming model in a manner that was accessible and understandable, emphasizing the balance between financial constraints and nutritional adequacy. The model's constraints were pivotal in ensuring that my weekly intake did not exceed the maximum sodium limit while meeting the necessary caloric and macronutrient thresholds. Part 3: Implementation and Computational Solution Upon running the model through AMPL, the solver's output initially indicated a single-food-item diet—50 servings of pizza—as the most cost-effective approach. This outcome clearly demonstrated the model's sensitivity to the cost parameter. Part 4: Initial Solution's Practicality The initial solution, while mathematically sound in minimizing costs, was not practically viable from a dietary standpoint. Consuming pizza exclusively, despite its cost-effectiveness, would not provide the variety of nutrients required for a healthy diet. Part 5: Adjusted Model for a Varied Diet To address the lack of diversity, I modified the model to include at least one serving of each selected food item. After re-solving, the solution indeed suggested a more diverse diet, albeit at an increased cost. The revised total cost for my diet plan was \$143.625 per week.

4. Conclusion

This exercise in applying linear programming to personal diet planning has illuminated the complexities involved in creating a cost-efficient yet nutritionally adequate diet. The iterative process of refining the model underscored the importance of incorporating a variety of foods to meet dietary guidelines. The final solution represents a balanced approach to diet optimization, considering both economic and nutritional values.

Appendices

Table 1: Cost Analysis

Food Item	Total Price	Weight per Container	Unit Price (per ounce)	Servings per Container	Cost per Serving
Almonds	$7.49	12 OZ	62.42 cents	12	$0.6242
Brown Rice	$3.69	8.8 OZ	41.93 cents	6.29 (approx.)	$0.5866
Greek Yogurt	$1.49	5.3 OZ	-	1	$1.49
Pizza	$11.49	19.1 OZ	-	4	$2.8725
Tofu Vegetables & Hash Brown	$6.99	9 OZ	77.67 cents	1	$6.99

Table 2: Nutritional Content Per Serving

Nutrient	Almonds (per oz)	Brown Rice (per 140g)	Greek Yogurt (per 150g)	Pizza (per ¼ pizza)	Tofu Vegetables & Hash Brown (per tray)
Sodium (mg)	0	0	55	630	790
Calories	160	240	110	270	420
Protein (g)	6	5	12	10	20
Vitamin D (mcg)	0	0	0	0.4	0
Calcium (mg)	70	10	10% DV*	180	150
Iron (mg)	1.1	0.9	0% DV*	1.9	3.6
Potassium (mg)	200	130			

amplpy-diet-example.py | diet | diet

File Edit View

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Based on AMPL Python API documentation:
# https://buildmedia.readthedocs.org/media/pdf/amplpy/latest/amplpy.pdf
import sys
import os
from amplpy import AMPL, Environment

def main(argc, argv):
    # Set up the AMPL environment
    ampl = AMPL(Environment('C:/Users/Lavieestbelle$1/AMPL'))

    # Set the solver to CPLEX
    ampl.setOption('solver', 'cplex')

    # Path to the model and data files
    model_directory = 'C:/Users/Lavieestbelle$1/Desktop/MSDS460/Assignment1/AMPL-WORK-2024/models/diet'

    # Read the model and data files
    ampl.read(os.path.join(model_directory, 'diet.mod'))
    ampl.read_data(os.path.join(model_directory, 'diet.dat'))

    # Solve the linear programming problem
    ampl.solve()

    # Print the total cost from the objective
    total_cost = ampl.get_objective('Total_Cost')
    print('Total cost for the diet plan is:', total_cost.value())

    # Print the optimal solution - servings of each food item
    servings = ampl.get_variable('Buy')
    df_servings = servings.get_values()
    print(df_servings)

    # Additional code for updating costs or other parameters can be added here

if __name__ == '__main__':
    try:
        main(len(sys.argv), sys.argv)
    except Exception as e:
        print(e)
        raise
```

Ln 36, Col 1 | 1,310 characters | 100% | Windows (CRLF) | UTF-8

ssibility: Good to go

amplpy-diet-example.py | diet | diet

File Edit View

```
data;

set NUTR := Sodium Energy Protein VitaminD Calcium Iron Potassium ;
set FOOD := Almonds Brown_Rice Greek_Yogurt Pizza Tofu_Vegetables_Hash_Brown ;

param:    cost  f_min  f_max :=
  Almonds                   0.6242   0    100
  Brown_Rice                0.7488   0    100
  Greek_Yogurt              1.49     0    100
  Pizza                     2.8725   0    100
  Tofu_Vegetables_Hash_Brown 6.99    0    100 ;

param:    n_min  n_max :=
   Sodium      0      35000
   Energy   14000    100000
   Protein   350     10000
   VitaminD  140      10000
   Calcium  9100    100000
   Iron      126     10000
   Potassium 32900 100000 ;

param amt (tr):
                   Sodium Energy Protein VitaminD Calcium Iron Potassium :=
   Almonds              0    1120    42       0     490    7.7   1400
   Brown_Rice           0    1680    35       0      70    6.3    910
   Greek_Yogurt        55     770    84       0      70    0      280
   Pizza              630    1890    70     2.8    1260   13.3   1820
   Tofu_Vegetables_Hash_Brown 790  2940   140      0   1050   25.2   5880 ;
```

Administrator: Anaconda Powershell Prompt

```
(ampl) PS C:\Users\Lavieestbelle$1\Desktop\MSDS460\Assignment1\AMPL-WORK-2024\models\diet> python .\amplpy-diet-example.
py
CPLEX 22.1.1.0: optimal solution; objective 143.625
0 dual simplex iterations (0 in phase I)
Total cost for the diet plan is: 143.625
   index0    |   Buy.val
 'Almonds'   |      0
'Brown_Rice' |      0
'Greek_Yogurt' |      0
  'Pizza'    |      50
'Tofu_Vegetables_Hash_Brown' |      0

(ampl) PS C:\Users\Lavieestbelle$1\Desktop\MSDS460\Assignment1\AMPL-WORK-2024\models\diet>
```

Here's how I updated the relevant section in my AMPL model:

param: cost f_min f_max :=

Almonds 0.6242 1 100

Brown_Rice 0.7488 1 100

Greek_Yogurt 1.49 1 100

Pizza 2.8725 1 100

Tofu_Vegetables_Hash_Brown 6.99 1 100 ;

By making this adjustment and solving the problem again, I anticipate that the total cost of my diet will increase. This is because I'm ensuring a variety of food items in my diet rather than opting for the single most cost-effective option. The new solution will reflect a more varied and likely more nutritionally balanced diet, as it will include at least one serving of each food item.

To determine how much more I'll need to spend on food each week, I will rerun the linear programming model with these new constraints. The difference between the new total cost and the previous one will tell me the additional amount. I expect this approach to yield a diet plan that is not only cost-aware but also richer in nutrients, aligning better with a healthy dietary pattern.