**Homework 11 Submission by Haritha Pulletikurti**

**Question 15.2**

In the videos, we saw the “diet problem”. (The diet problem is one of the first large-scale optimization

problems to be studied in practice. Back in the 1930’s and 40’s, the Army wanted to meet the nutritional

requirements of its soldiers while minimizing the cost.) In this homework you get to solve a diet problem with real data. The data is given in the file diet.xls.

1. Formulate an optimization model (a linear program) to find the cheapest diet that satisfies the maximum and minimum daily nutrition constraints, and solve it using PuLP. Turn in your code and the solution. (The optimal solution should be a diet of air-popped popcorn, poached eggs, oranges, raw iceberg lettuce, raw celery, and frozen broccoli. UGH!)
2. Please add to your model the following constraints (which might require adding more variables) and solve the new model:
3. If a food is selected, then a minimum of 1*/*10 serving must be chosen. (Hint: now you will need two variables for each food *i*: whether it is chosen, and how much is part of the diet. You’ll also need to write a constraint to link them.)
4. Many people dislike celery and frozen broccoli. So at most one, but not both, can be selected.
5. To get day-to-day variety in protein, at least 3 kinds of meat/poultry/fish/eggs must be selected. [If something is ambiguous (e.g., should bean-and-bacon soup be considered meat?), just call it whatever you think is appropriate – I want you to learn how to write this type of constraint, but I don’t really care whether we agree on how to classify foods!]

If you want to see what a more full-sized problem would look like, try solving your models for the file diet\_large.xls, which is a low-cholesterol diet model (rather than minimizing cost, the goal is to minimize cholesterol intake). I don’t know anyone who’d want to eat this diet – the optimal solution includes dried chrysanthemum garland, raw beluga whale flipper, freeze-dried parsley, etc. – which shows why it’s necessary to add additional constraints beyond the basic ones we saw in the video!

[**Note**: there are many optimal solutions, all with zero cholesterol, so you might get a different one. It probably won’t be much more appetizing than mine.]

**Answer:**

The Diet Optimization problem listed above is approached in two parts.

# Part 1: Optimization Problem to minimize the Cost of the food.

# Data:

= amount of nutrient per unit of food

= minimum amount of nutrient required

= maximum amount of nutrient required

= cost per unit of food

# Variables: = amount of food eaten

= binary variable indicating if food “i” is used.

Constraints:

for each nutrient (food has at least minimum amount of each nutrient) for each nutrient (food has no more than maximum amount of each nutrient)

for each food (cannot eat negative amounts)

Objective Function:

# Minimize (minimize total cost)

# Part 1: Optimization Problem to minimize the Cost of the food.

Please see the Homework11\_Part1.py and Homework11\_Part1.html for the code and explanation with results to solve the optimization problem stated in Question 15.2 – 1, 2- a, b, c.

**To minimize the cost, I used:** “OptimizationProblem = LpProblem('PuLPTutorial', LpMinimize)”

**Variables:**

1. xi = amount of food in the diet and x cannot go below 0.

x=LpVariable.dicts("Amounts", foods, 0)

1. yi = Binary variable indicating if the food i is used (0 or 1 are the possibilities)

y=LpVariable.dicts("Chosen",foods,0,1,"Binary")

**Constraints:**

1. Added Minimum calory intake Constraints to all the food

for i in range (0,11):

dotproductM = pulp.lpSum([M[i][j] \* x[j] for j in foods])

minumum\_Condition = minimum[i] <= + dotproductM

OptimizationProblem += minumum\_Condition

1. Added Maximum calory intake Constraints to all the food

for i in range(0,11):

dotproductM = pulp.lpSum([M[i][j] \* x[j] for j in foods])

maximum\_Condition = maximum[i] >= + dotproductM

OptimizationProblem += maximum\_Condition

3. If a food is selected, then a minimum of 1/10 serving must be chosen.

The food also needs to be less than a large amount. I have arbitrarily chosen 100000

for i in foods:

OptimizationProblem += x[i] <= 100000 \* y[i]

OptimizationProblem += x[i] >= 0.1 \* y[i]

4. Many people dislike celery and frozen broccoli. So, at most one, but not both, can be selected.

OptimizationProblem +=y['Frozen Broccoli'] + y['Celery, Raw'] <= 1

5.To get day-to-day variety in protein, at least 3 kinds of meat/poultry/fish/eggs must be selected.

OptimizationProblem += y['Roasted Chicken'] + y['Poached Eggs'] + y['Frankfurter, Beef'] +

y['Scrambled Eggs'] + y['Hamburger W/Toppings'] + y['Pizza W/Pepperoni']+

y['Bologna,Turkey'] + y['Kielbasa,Prk'] +y['White Tuna in Water'] +y['Sardines in Oil'] +

y['Ham,Sliced,Extralean'] + y['Hotdog, Plain'] + y['Pork’] +y['Chicknoodl Soup']+

y ['Splt Pea&Hamsoup'] + y['Vegetbeef Soup'] + y['Beanbacn Soup,W/Watr']>= 3

**Objective Function:** Minimize the cost of the food

OptimizationProblem += lpSum([cost[i] \* x[i] for i in foods])

**Optimization Solution is:**

42.399358units of Amounts\_Celery,\_Raw

0.1units of Amounts\_Kielbasa,Prk

82.802586units of Amounts\_Lettuce,Iceberg,Raw

3.0771841units of Amounts\_Oranges

1.9429716units of Amounts\_Peanut\_Butter

0.1units of Amounts\_Poached\_Eggs

13.223294units of Amounts\_Popcorn, Air\_Popped

0.1units of Amounts\_Scrambled\_Eggs

Total cost of food = $4.51

**Part 2- Optimization of the Cholesterol intake:**

Please see the Homework11\_Part2.py and Homework11\_Part2.html for the code and explanation with results to solve the optimization problem stated in Question 15.2 – last paragraph.

Data: diet\_large.xls

**To minimize the cholesterol intake, I used:**

“OptimizationProblem = LpProblem('Diet Optimization', LpMinimize)”

**Variables:** xi = LpVariable.dicts("Foods", foods, 0)

**Constraints:** Added Minimum and Maximum calory intake Constraints to all the food

for i in range(0,Nutrientcount):

if (not np.isnan(minimumVal[0][i+1])) and (not np.isnan(maximumVal[0][i+1])):

OptimizationProblem +=

lpSum([N[i][j] \* xi[j] for j in foods]) >= minimumVal[0][i+1], 'min nutrient ' + Nutrients[i+1]

OptimizationProblem +=

lpSum([N[i][j] \* xi[j] for j in foods]) <= maximumVal[0][i+1], 'max nutrient ' + Nutrients[i+1]

**Objective Function:** Minimize the Cholesterol intake.

OptimizationProblem += lpSum([cost[i] \* xi[i] for i in foods])

**Optimal Solution:**

0.059863415 units of Beans,\_adzuki,\_mature\_seeds,\_raw

0.069514608 units of Broccoli\_raab,\_raw

0.42866218 units of Cocoa\_mix,\_no\_sugar\_added,\_powder

0.14694398 units of Egg,\_white,\_dried,\_flakes,\_glucose\_reduced

0.73805891 units of Infant\_formula,\_MEAD\_JOHNSON,\_ENFAMIL,\_NUTRAMIGEN,\_with\_iron,\_p

0.4258564 units of Infant\_formula,\_NESTLE,\_GOOD\_START\_ESSENTIALS\_\_SOY,\_\_with\_iron,

0.050114149 units of Infant\_formula,\_ROSS,\_ISOMIL,\_with\_iron,\_powder,\_not\_reconstitu

0.15033656 units of Margarine\_like\_spread,\_approximately\_60%\_fat,\_tub,\_soybean\_(hyd

0.25918767 units of Mung\_beans,\_mature\_seeds,\_raw

0.18052856 units of Nuts,\_mixed\_nuts,\_dry\_roasted,\_with\_peanuts,\_with\_salt\_added

1.184482 units of Oil,\_vegetable,\_sunflower,\_linoleic,\_(hydrogenated)

0.10375187 units of Seeds,\_sunflower\_seed\_kernels,\_dry\_roasted,\_with\_salt\_added

0.031866196 units of Snacks,\_potato\_chips,\_fat\_free,\_made\_with\_olestra

0.070710308 units of Spices,\_paprika

0.55106575 units of Tomatoes,\_sun\_dried

9999.6864 units of Water,\_bottled,\_non\_carbonated,\_CALISTOGA

Total Cholestrol = 0.000000