# MCEN 5125

Homework 4

Hanwen Zhao

April 26, 2018

# 1 Promblem 1: Diet Problem

The original diet problem data spreadsheet provides 64 different items with their corresponding Price, Calories, Cholesterol, Total Fat, Sodium, Carbohydrates, Dietary Fiber, Protein, Vitamin A, Vitamin C, Calcium and Iron nutrients values per serving.
The second sheet provides the minimum and maximum value of those nutrients per day.
To be more clear on this problem, we have 11 nutrients and the price per serving as our variables. In addition to the following constraints

- Each nutrients value for all food selections must greater than the minimum value.
- Each nutrients value for all food selections must smaller than the maximum value.
- The quantity for all food selections must equal or greater than zero.

Thus, we can conclude the problem as an LP in Inequality form:

$$minimize\ c^T x$$
$$subject\ to\ Ax \leqslant b$$

where c is the price per serving, x is the solution for the food selection with corresponding serving size. A is the variable matrix which contains all the nutrients information. In addition the b is the constrains where contains the minimum and maximum values for daily nutrients.

## 1.1 Part I

The first portion of the A matrix is the constrains equations which defines the minimum daily nutrients values needed.

$$Calories_{Item1} * ServingSize_{Item1} + Calories_{Item2} * ServingSize_{Item2} + \cdots + Calories_{Item64} * ServingSize_{Item64} \geqslant Calories_{min}$$
$$Cholesterol_{Item1} * ServingSize_{Item1} + Cholesterol_{Item2} * ServingSize_{Item2} + \cdots + Cholesterol_{Item64} * ServingSize_{Item64} \geqslant Cholesterol_{min}$$
$$\cdots$$
$$Iron_{Item1} * ServingSize_{Item1} + Iron_{Item2} * ServingSize_{Item2} + \cdots + Iron_{Item64} * ServingSize_{Item64} \geqslant Iron_{min}$$

It results the following matrix:

$$A1 = \begin{bmatrix} Calories_{Item1} & Calories_{Item2} & Calories_{Item3} & \dots & Calories_{Item64} \\ Cholesterol_{Item1} & Cholesterol_{Item2} & Cholesterol_{Item3} & \dots & Cholesterol_{Item64} \\ \dots & \dots & \dots & \dots & \dots \\ Iron_{Item1} & Iron_{Item2} & Iron_{Item3} & \dots & Iron_{Item64} \end{bmatrix}$$

and

$$b1 = \begin{bmatrix} Calories_{min} \\ Cholesterol_{min} \\ \dots \\ Iron_{Imin} \end{bmatrix}$$

## 1.2 Part II

The second portion of the A matrix is the constrains equations which defines the maximum daily nutrients values needed.

$$Calories_{Item1} * ServingSize_{Item1} + Calories_{Item2} * ServingSize_{Item2} + \cdots + Calories_{Item64} * ServingSize_{Item64} \leqslant Calories_{max}$$
$$Cholesterol_{Item1} * ServingSize_{Item1} + Cholesterol_{Item2} * ServingSize_{Item2} + \cdots + Cholesterol_{Item64} * ServingSize_{Item64} \leqslant Cholesterol_{max}$$
$$\ldots$$
$$Iron_{Item1} * ServingSize_{Item1} + Iron_{Item2} * ServingSize_{Item2} + \cdots + Iron_{Item64} * ServingSize_{Item64} \leqslant Iron_{max}$$

It results the following matrix:

$$A2 = \begin{bmatrix} Calories_{Item1} & Calories_{Item2} & Calories_{Item3} & \ldots & Calories_{Item64} \\ Cholesterol_{Item1} & Cholesterol_{Item2} & Cholesterol_{Item3} & \ldots & Cholesterol_{Item64} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ Iron_{Item1} & Iron_{Item2} & Iron_{Item3} & \ldots & Iron_{Item64} \end{bmatrix}$$

and

$$b2 = \begin{bmatrix} Calories_{max} \\ Cholesterol_{max} \\ \ldots \\ Iron_{max} \end{bmatrix}$$

## 1.3 Part III

The last portion of the A matrix is the constrains equations which defines any serving size must equal or greater than 0.

$$ServingSize_{Item1} \geqslant 0$$
$$ServingSize_{Item2} \geqslant 0$$
$$\ldots$$
$$ServingSize_{Item64} \geqslant 0$$

It results the following matrix:

$$A3 = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

and

$$b3 = \begin{bmatrix} 0 \\ 0 \\ . \\ 0 \end{bmatrix}$$

## 1.4 Solution

In MATLAB, the default linprog solves:

$$X = linprog(f, A, b)$$
$$minimize\ f'x$$
$$subject\ to\ Ax \leqslant b$$

In order to solve the problem with matrix we have now, we need to form:

$$A = \begin{bmatrix} -A1 \\ A2 \\ -A3 \end{bmatrix}$$

and

$$b = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix}$$

and

$$f = \begin{bmatrix} Price_{Item1} \\ Price_{Item2} \\ .......... \\ Price_{Item64} \end{bmatrix}$$

It gives us the following solution:

- 0.24 serving of Carrots,Raw
- 3.54 serving of Potatoes, Baked
- 2.17 serving of Skim Milk
- 3.60 serving of Peanut Butter
- 4.82 serving of Popcorn,Air-Popped

which satisfies all the nutrients requirements:

- 2000.00 cal Calories
- 9.54 mg Cholesterol
- 65.00 g Total Fat
- 897.22 g Sodium
- 300.00 g Carbohydrates
- 75.59 g Protein
- 5000.00 IU Vitamin A
- 61.71 IU Vitamin C
- 800.00 mg Calcium
- 21.55 mg Iron

As the result, the total cost is 0.96 dollars.

### 1.5 Play with the Problem

However the diet looks terrible to me, so we can modify the result so that the diet meeting my preference. One way to achieve is to modify the price for the item, increase the price for the item I don't like so that the algorithm will not pick them since it's trying to minimum the total cost.

After the modification, I get the following result:

- 3.89 serving of Potatoes, Baked
- 3.60 serving of Oranges
- 2.49 serving of Chocolate Chip Cookies
- 0.79 serving of Scrambled Eggs
- 2.10 serving of Taco
- 0.34 serving of Splt Pea Hamsoup

which satisfies all the nutrients requirements:

- 2000.00 cal Calories
- 300.00 mg Cholesterol
- 63.04 g Total Fat
- 2347.49 g Sodium
- 300.00 g Carbohydrates
- 25.00 g Dietary Fiber
- 73.45 g Protein
- 5000.00 IU Vitamin A
- 318.43 IU Vitamin C
- 800.00 mg Calcium
- 24.39 mg Iron

Although the final cost increased to 2.40 dollars, but it is still super cheap, and now the diet looks more eatable.

# 2 Problem 2: Scaffolding Problem

We can start this problem by doing the statics analysis, we can obtain the following equations:

$$\begin{aligned} T_3 + T_4 - W_2 &= 0 \\ -4T_3 + 4T_4 &= 0 \\ T_1 + T_2 - T_3 - W_1 &= 0 \\ -8T_1 + 8T_2 - 4T_3 &= 0 \end{aligned}$$

In addition, each rope has the maximum tension it can take, so the tension on each rope mush smaller than the maximum tension. Since we are solving this problem in Equality Form, we need to introduce some slack, now we have the following equations:

$$\begin{aligned} T_1 + S_1 &= 120 \\ T_2 + S_2 &= 160 \\ T_3 + S_3 &= 100 \\ T_4 + S_4 &= 100 \end{aligned}$$

where S are slack variables.
Also for all tension on ropes must greater than 0, we can do this in linprog by setting the lower bound.

## 2.1 Building Matrix

Now we can start building our matrix:

$$x = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ W_1 \\ W_2 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}$$

From statics equations, we have:

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & -4 & 4 & 0 & 0 \\ 1 & 1 & -1 & 0 & -1 & 0 \\ -8 & 8 & -4 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

For maximum tension constrains, we have:

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

finally,

$$A = \begin{bmatrix} A1 \\ A2 \end{bmatrix}$$

and b equal to:

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 120 \\ 160 \\ 100 \\ 100 \end{bmatrix}$$

We are solving for maximum of $W_1 + W_2$,

$$f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

As the result, we have:

$$W_1 = 170$$
$$W_2 = 200$$

# 3 Appendix

## 3.1 MATLAB Code: Diet Problem

```
% MCEN 5125
% Homework 4
% Hanwen Zhao
% MEID: 650-703

clear all
close all
clc

% read the excel data
[~,~,food] = xlsread('DietProblemData.xlsx',1);
[~,~,nutrient] = xlsread('DietProblemData.xlsx',2);
% form the first third A matrix
% A1x >= bmin
% A1 needs to be negative since the default in linprog is <=
A1 = -1 * transpose(cell2mat(food(2:end,2:end-1)));
% form the second thrid A matrix
% A2x <= bmax
A2 = transpose(cell2mat(food(2:end,2:end-1)));
% form the last portion of the A matrix
% Ix >= 0
% Again the A3 needs to be negative
A3 = -1 * eye(length(A1));
```

```
% the f is the price of each item, we want to mininmize this
f = transpose(cell2mat(food(2:end,end)));
% the first part of the b is the minimum value of nutrients
% the b1 needs to be negative
b1 = -1 * cell2mat(nutrient(2:end,3:end-1));
% the second part of the b is the maximum value of nutrients
b2 = cell2mat(nutrient(2:end,end));
% the third part of the b matrix are zeros
b3 = zeros(length(A1),1);
% now we can form the final matrix
A = [A1;A2;A3];
b = [b1;b2;b3];
% call linprog
result = linprog(f,A,b);
% get the index and quantity of the food selection
index = find(result~=0)+1;
quantity = result(index-1);
% print the result
fprintf('In order to meet the nutrients requirements, the following items are selected: \n')
for i=1:length(index)
    fprintf('%4.2f serving of %s \n', quantity(i), food{index(i),1})
end
% calculate the final nutrient values
nutrientResult = -A*result;
nutrientResult = nutrientResult(1:11);
% print out the values
fprintf('\n\nThe final nutrient values contains \n')
fprintf('%4.2f cal Calories \n', nutrientResult(1))
fprintf('%4.2f mg Cholesterol \n', nutrientResult(2))
fprintf('%4.2f g Total_Fat \n', nutrientResult(3))
fprintf('%4.2f g Sodium \n', nutrientResult(4))
fprintf('%4.2f g Carbohydrates \n', nutrientResult(5))
fprintf('%4.2f g Dietary_Fiber \n', nutrientResult(6))
fprintf('%4.2f g Protein \n', nutrientResult(7))
fprintf('%4.2f IU Vitamin A \n', nutrientResult(8))
fprintf('%4.2f IU Vitamin C \n', nutrientResult(9))
fprintf('%4.2f mg Calcium \n', nutrientResult(10))
fprintf('%4.2f mg Iron \n', nutrientResult(11))

fprintf('\n\nThe total cost is %4.2f dollars.\n', f*result)
```

### 3.2 MATLAB Code: Scaffolding Problem

```
% MCEN 5125
% Homework 4
% Hanwen Zhao
% MEID: 650-703

clear all
close all
clc
% Building constrain matrix from statics equations
A1 = [0,0,1,1,0,-1;
      0,0,-4,4,0,0;
      1,1,-1,0,-1,0;
```

```
      -8,8,-4,0,0,0];
A1 = [A1,zeros(4,4)];
% Add slack varibales so that tension is smaller than the maximum
A2 = [eye(4,6),eye(4,4)];
% combine the previous problem
A = [A1;A2];
% statics equations have 0 on rhs
b1 = zeros(4,1);
% define the maximum tensions
b2 = [120;160;100;100];
b = [b1;b2];
% -1 for W1 and W2 so we are minimize
f = [zeros(4,1);-1;-1;zeros(4,1)];
% call linprog and set lower bound to 0
result = linprog(f,[],[],A,b,zeros(10,1),[]);
W1 = result(5)
W2 = result(6)
```