MEC 320 Programming Assignment II

114446084 Ching Wei Huang

Introduction:

In this assignment, we want to find the maximum profit of the diary plant show below.

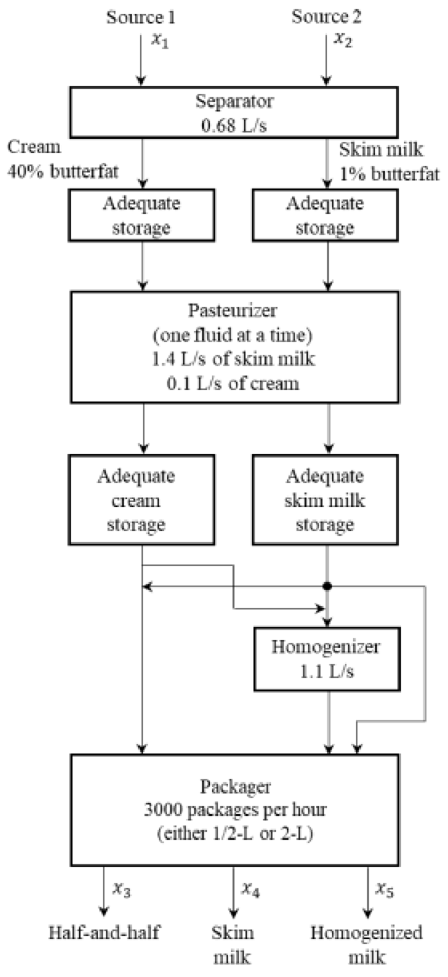


Figure 1 Flow diagram for a dairy plant

The plant takes 2 input ingredients x_1 and x_2 and produce 3 final products: half-and-half, skim milk, and homogenized milk. The cost and selling price for all ingredients and products are shown in the table below

Item	Designation, L/day	Sale or purchase cost	Butterfat content, volume %
Source 1	x_1	\$0.23 per liter	4.0
Source 2	x_2	0.24 per liter	4.5
Half-and-half	x_3	0.48 per half-liter	$\geq$ 10.0
Skim milk	x_4	0.60 per two-liter	$\geq$ 1.0
Homogenized milk	x_5	0.68 per two-liter	$\geq$ 3.0

The problem is solved using linear programming.

Methodology:

a The objective function Z(x) is

$$Z(x) = -0.23x_1 - 0.24x_2 + 2 * 0.48x_3 + 0.5 * 0.60x_4 + 0.5 * 0.68x_5 \qquad (1)$$

Z(x) implements the profit for the process, the goal of this linear programming problem is to maximize Z(x)

b The constraints of the problems are

$$x_1 + x_2 \le 19584 \qquad (2)$$

$$x_1 + 1.08x_2 \le 20150 \qquad (3)$$

$$x_5 \le 31680 \qquad (4)$$

$$4x_3 + x_4 + x_5 \le 48000 \qquad (5)$$

$$-4x_1 - 4.5x_2 + 10x_3 + x_4 + 3x_5 \le 0 \qquad (6)$$

$$-x_1 - x_2 + x_3 + x_4 + x_5 = 0 \qquad (7)$$

Constraint 2 to 6 comes from the assignment instruction. Constraint 7 comes from the conservation of mass, which the amount of total input ingredients x_1 and x_2 should equal to total output ingredients x_3 x_4 and x_5.

In terms of simplex method, the problem can be written as

```
A=[1 1 0 0 0 1 0 0 0 0;
   1 1.08 0 0 0 0 1 0 0 0;
  0 0 0 0 1 0 0 1 0 0;
  0 0 4 1 1 0 0 0 1 0;
  -4 -4.5 10 1 3 0 0 0 0 1;
   1 1 -1 -1 -1 0 0 0 0 0]
b=[19584;20150;31680;48000;0;0];
C=[-0.23;-0.24;0.96;0.30;0.34;0;0;0;0;0];
X=[x1 x2 x3 x4 x5 s1 s2 s3 s4 s5]
```

Where

$$Z(x) = C^T X$$

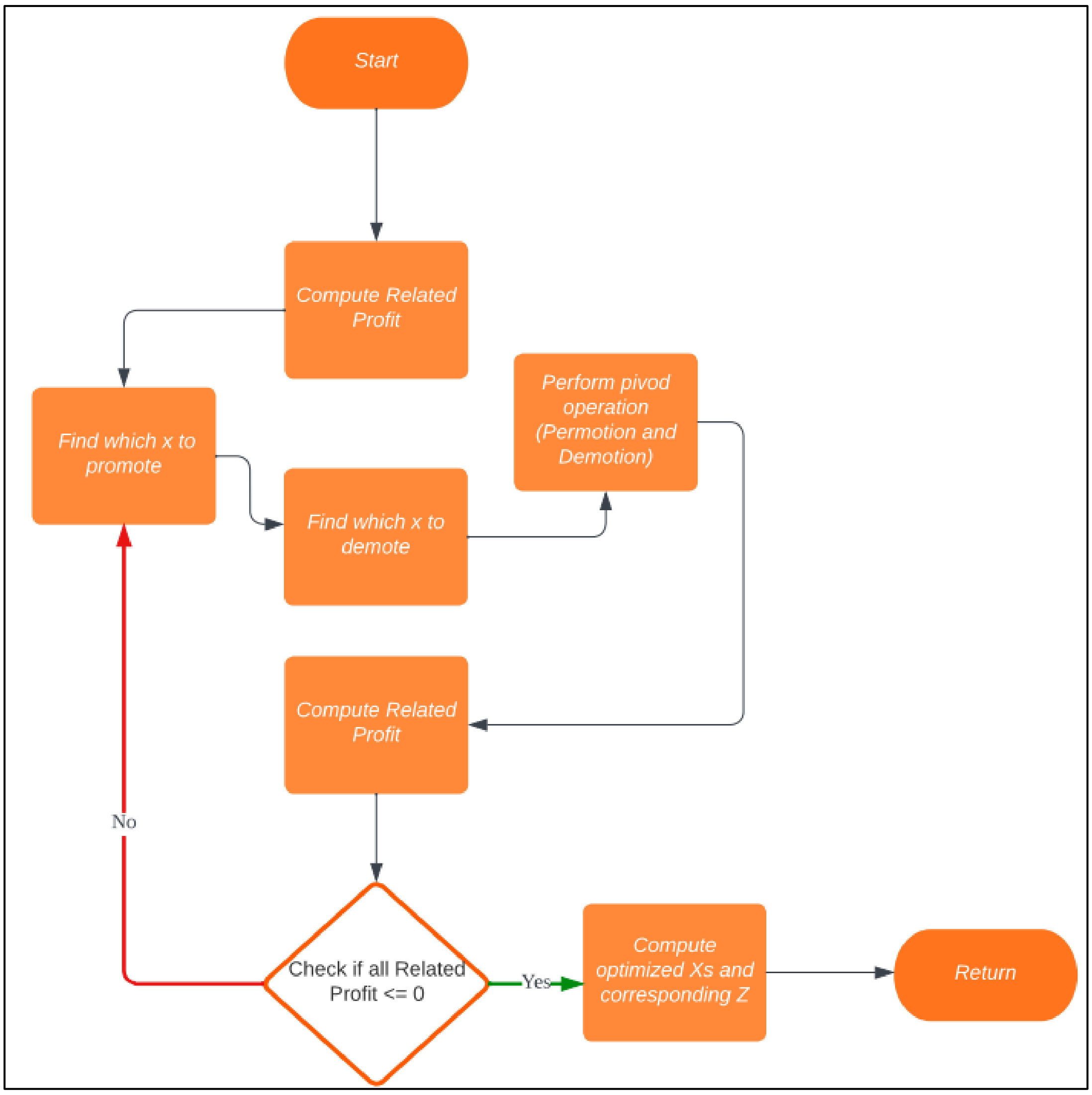
$$AX = b, \qquad X, b \ge 0$$

c The function to solve the linear programming problem is shown as

```
function [optimizedXs, optimizedFunVal] = linearProg(C,A,b)
```

It takes 3 inputs matrix C, A and b which represents the coefficients of constraints, the right side of constraints, and the coefficients of the objective function respectively in the Simplex method in linear programming. The function returns the optimal solution as a matrix "`optimizedXs`" the optimal value “`optimizedFunVal`” of the objective function.

The computational steps in the “`linearProg(C,A,b)`” function follows the flow chart below.



Start
Compute Related Profit
Find which x to promote
Find which x to demote
Perform pivod operation (Permotion and Demotion)
Compute Related Profit
Check if all Related Profit <= 0
No
Yes
Compute optimized Xs and corresponding Z
Return

Figure 1 Flow Chart of the “linearProg” Function

A helper method calls “`pivodOperation`”

```
Function [pivotted_matrix, pivotted_b]
=pivodOperation(A,b,targetRow,targetCol)
```

is written in the program to support the computation. The method takes 2 matrixes A and b and 2 integer `targetRow and targetCol` as input. In the end, it returns the matrixes “`pivotted_matrix”, and “pivotted_b”` that make through the pivot operation (Guess-Jordan) in the augmented matrix [A|b] at the `(targetRow,targetCol)` element.

The source code is written in MATLAB and provided in the appendix below.

Results and Discussion:

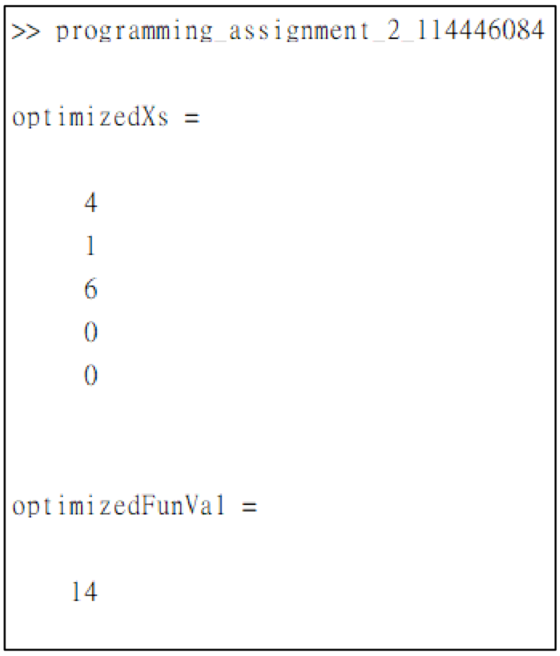
d The example 1 problem used in class for Linear Programming is used to test the code. With input

```
A=[-1 2 1 0 0;3 2 0 1 0; 1 -1 0 0 1];
b=[4;14;3];
C=[3;2;0;0;0];
```

The return results of the line of code:

```
[optimizedXs, optimizedFunVal]=linearProg(C,A,b)
```

are show in figure 2.



```
>> programming_assignment_2_114446084

optimizedXs =


     4
     1
     6
     0
     0



optimizedFunVal =


    14
```

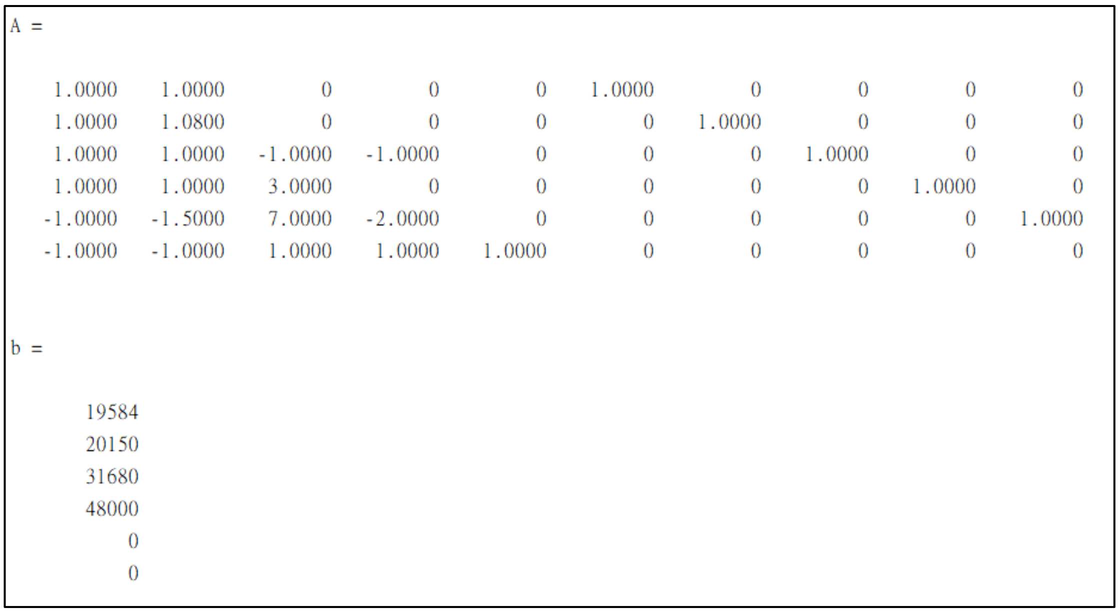
Figure 2 Optimize Solution of the Example 1 Problem Used in Class for Linear Programming Solved by the "linearProg" Function

The solution calculated by the code is same as the solution calculated in class. The "programming_assignment_2_114446084.m" file provided separately shows step by step computation of this result, which is identical to the steps in the slides shown in class.

e The solution of the assigned problem is computed in this section. To form a proper Canonical form, the input matrix A and b should be tweak into

```
[A, b]=pivodOperation(A,b,6,5)
```

where the A and b at the right side is same as shown in section b. The tweaked A and b are



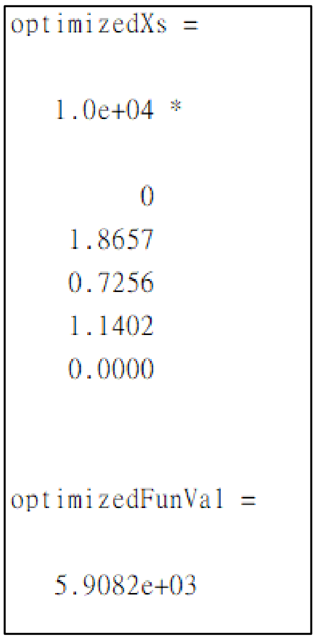
```
A =

    1.0000    1.0000         0         0         0    1.0000         0         0         0         0
    1.0000    1.0800         0         0         0         0    1.0000         0         0         0
    1.0000    1.0000   -1.0000   -1.0000         0         0         0    1.0000         0         0
    1.0000    1.0000    3.0000         0         0         0         0         0    1.0000         0
   -1.0000   -1.5000    7.0000   -2.0000         0         0         0         0         0    1.0000
   -1.0000   -1.0000    1.0000    1.0000    1.0000         0         0         0         0         0


b =

       19584
       20150
       31680
       48000
           0
           0
```

Figure 3 Tweaked A and b to form a proper Canonical form in the input of the “linearProg” Function

With the inputs the return results of the line of code:

```
[optimizedXs, optimizedFunVal]=linearProg(C,A,b)
```

are show in figure 4.



```
optimizedXs =

   1.0e+04 *

         0
    1.8657
    0.7256
    1.1402
    0.0000


optimizedFunVal =

   5.9082e+03
```

Figure 2 Optimize Solution of the Assigned Problem

The solution yields that when

- $\mathit{Input}\ \text{Source}\ 1 = x_1 = 0$ (L/day)
- $\mathit{Input}\ \text{Source}\ 2 = x_2 = 18657$ (L/day)
- $\text{Half and half} = x_3 = 7256$ (L/day)
- $\text{Skim milk} = x_4 = 11402$ (L/day)

- Homogenized milk $= x_5 = 0$ (L/day)

the profit is maximum, and the factory could earn at most \$5908/day.

Compare the result with the constrains,

$$x_1 + x_2 = 18657 \le 19584 \quad (8)$$

$$x_1 + 1.08x_2 = 20149.56 \le 20150 \quad (9)$$

$$x_5 = 0 \le 31680 \quad (10)$$

$$4x_3 + x_4 + x_5 = 40426 \le 48000 \quad (11)$$

$$-4x_1 - 4.5x_2 + 10x_3 + x_4 + 3x_5 = 5.5 \nleq 0 \quad (12)$$

$$-x_1 - x_2 + x_3 + x_4 + x_5 = 0 = 0 \quad (13)$$

Result show in equation 12 slightly violates constraint; The floating-point error of the computer could explain this small error.

<u>Conclusion:</u>

There are some minor bugs in my written MATLAB `linearProg` function. The user has to input with a proper Canonical form for matrix A or my function would crash. Secondly, the while loop for checking related profit has condition: all related profit<0.0001 instead of all related profit$\leqq$0 because I cannot find a better way to deal with round off errors in my calculation. Lastly, I want to make some complain about this assignment. I had a lot of works and even another programming assignment for another class due this week, so I choose do and submit this assignment early even I know my code has some bugs because it would waste me too much time to fix it. I felt terrible when I know the grading criteria and a sample code was changed and given 1 day before the deadline. It makes my effort and time management look useless. Therefore, instead of the sample code, I decided to hand out my original version of code which I put my most effort on.

Appendix:

```
% z=C'X (with slack), AX=b (with slack), defult all X>=0
function [optimizedXs, optimizedFunVal]=linearProg(C,A,b)
    Crow=C';
   Cb=Crow(size(C)-size(b)+1:size(C))';


       % Check Related Profit, not<=0 since floating point error
    while(~all(Crow(:)<=0.0001))
        % Find which x to promote
        [~, xPromote]=max(Crow(:));


        % Find which x to demote
        Ratio=b./A(:,xPromote);

        [~, xDemote]=min(abs(Ratio));
        Cb(xDemote)=C(xPromote);


        % Perform pivod operation
        [A, b]=pivodOperation(A,b,xDemote,xPromote);

        %Compute Related Profit
        Crow=zeros(size(Crow));
        for ii=1:size(Crow,2)
            if(sum(A(:,ii).^2)==1)
                Crow(ii)=0;
            else
                Crow(ii)=C(ii)-sum(Cb.*A(:,ii));
            end
        end

    end

    %Compute optimized Xs and corresponding Z
    Xs=zeros(size(C));
    temp=A.^2;
    for ii=1:size(A,2)
        if(sum(temp(:,ii))==1)
            [~,xposition]=max(A);
            Xs(ii)=b(xposition(ii));
        end
    end
    optimizedXs=Xs;
    optimizedFunVal=Cb'*b;

end



function [pivotted_matrix, pivotted_b]=
pivodOperation(A,b,targetRow,targetCol)
    M=[A b];
    C=M(targetRow,:);

    for ii=1:targetRow-1
        factor=M(ii,targetCol)/C(targetCol);
        for jj=1:size(M,2)
            M(ii,jj)=M(ii,jj)-factor*C(jj);
        end
    end
    if((targetRow+1)<=size(M,1))
        for ii=targetRow+1:size(M,1)
            factor=M(ii,targetCol)/C(targetCol);
            for jj=1:size(M,2)
                M(ii,jj)=M(ii,jj)-factor*C(jj);
            end
        end
    end

    M(targetRow,:)=M(targetRow,:)./M(targetRow,targetCol);

    pivotted_matrix=M(:,1:size(M,2)-1);
    pivotted_b=M(:,size(M,2));
end
```