Students: Group 11 (Shawn Hillyer, Jesse Thoren, Jason Goldfine-Middleton)
Assignment: Project 3
Due date: 11/13/2016

Problem 1

Part A

Determine the number of refrigerators to be shipped plants to warehouses and then warehouses to retailers to minimize the cost.

i. Formulate the problem as a linear program with an objective function and all constraints.
ii. Determine the optimal solution for the linear program using any software you want. Include a copy of the code/file in the report.
iii. What are the optimal shipping routes and minimum cost.

Part A-i

Define Variables:

Let p be the Plants {P1, P2, P3, P4}
Let w be the Warehouses {W1, W2, W3}
Let r be the Retailers {R1, R2, R3, R4, R5, R6, R7}
Let X_{pw} be the units to ship from p to w
Let Y_{wr} be the units to ship from w to r
Let S_p be the supply available at each plant p
Let D_r be the demand at each retailer r.

Objective: Minimize the cost of shipping the products from the plants to the warehouses and warehouses to the retailers.

MINIMIZE $10X_{11} + 15X_{12} + 11X_{21} + 8X_{22} + 13X_{31} + 8X_{32} + 9X_{33} + 14X_{42} + 8X_{43} + 5Y_{11} + 6Y_{12} + 7Y_{13} + 10Y_{14} + 12Y_{23} + 8Y_{24} + 10Y_{25} + 14Y_{26} + 14Y_{34} + 12Y_{35} + 12Y_{36} + 6Y_{37}$

Constaints:
s.t.:

// Each Plant cannot ship more than it has
$X_{11} + X_{12} \leq 150$ //p1 supply
$X_{21} + X_{22} \leq 450$ //p2 supply
$X_{31} + X_{32} + X_{33} \leq 250$ //p3 supply
$X_{42} + X_{43} \leq 150$ //p4 supply

// Each retailer receives exactly what its demand is
$Y_{11} = 100$ // Retailer 1 demand
$Y_{12} = 150$ // Retailer 2 demand
$Y_{13} + Y_{23} = 100$ // Retailer 3 demand
$Y_{14} + Y_{24} + Y_{34} = 200$ // Retailer 4 demand
$Y_{25} + Y_{35} = 200$ // Retailer 5 demand
$Y_{26} + Y_{36} = 150$ // Retailer 6 demand
$Y_{37} = 100$ // Retailer 7 demand

// Units shipped from each warehouse must be <= amount shipped to the warehouse

$$Y_{11} + Y_{12} + Y_{13} + Y_{14} - X_{11} - X_{21} - X_{31} \leq 0$$
$$Y_{23} + Y_{24} + Y_{25} + Y_{26} - X_{12} - X_{22} - X_{32} - X_{42} \leq 0$$
$$Y_{34} + Y_{35} + Y_{36} + Y_{37} - X_{33} - X_{43} \leq 0$$

Non-negativity constraints:

X_{pw} and $Y_{wr} \geq 0$

Part A-ii

The optimal solution was generated using the above formulas translated into a LINDO program.

Code is in file `CS325Project3Problem1a.ltx` as well as below. The output from LINDO follows the code.

```
MINIMIZE 10X11 + 15X12 + 11X21 + 8X22 + 13X31 + 8X32 + 9X33 + 14X42 + 8X43 + 5Y11 +
6Y12 + 7Y13 + 10Y14 + 12Y23 + 8Y24 + 10Y25 + 14Y26 + 14Y34 + 12Y35 + 12Y36 + 6Y37
ST
	X11 + X12 <= 150
	X21 + X22 <= 450
	X31 + X32 + X33 <= 250
	X42 + X43 <= 150
	Y11 = 100
	Y12 = 150
	Y13 + Y23 = 100
	Y14 + Y24 + Y34 = 200
	Y25 + Y35 = 200
	Y26 + Y36 = 150
	Y37 = 100
	Y11 + Y12 + Y13 + Y14-X11-X21-X31 <= 0
	Y23 + Y24 + Y25 + Y26-X12-X22-X32-X42 <= 0
	Y34 + Y35 + Y36 + Y37-X33-X43 <= 0
```

Results:

```
 LP OPTIMUM FOUND AT STEP     10

        OBJECTIVE FUNCTION VALUE

        1)      17100.00

  VARIABLE        VALUE          REDUCED COST
       X11       150.000000          0.000000
       X12         0.000000          8.000000
       X21       200.000000          0.000000
       X22       250.000000          0.000000
       X31         0.000000          2.000000
       X32       150.000000          0.000000
       X33       100.000000          0.000000
       X42         0.000000          7.000000
       X43       150.000000          0.000000
       Y11       100.000000          0.000000
```

```
       Y12       150.000000          0.000000
       Y13       100.000000          0.000000
       Y14         0.000000          5.000000
       Y23         0.000000          2.000000
       Y24       200.000000          0.000000
       Y25       200.000000          0.000000
       Y26         0.000000          1.000000
       Y34         0.000000          7.000000
       Y35         0.000000          3.000000
       Y36       150.000000          0.000000
       Y37       100.000000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          1.000000
        3)         0.000000          0.000000
        4)         0.000000          0.000000
        5)         0.000000          1.000000
        6)         0.000000        -16.000000
        7)         0.000000        -17.000000
        8)         0.000000        -18.000000
        9)         0.000000        -16.000000
       10)         0.000000        -18.000000
       11)         0.000000        -21.000000
       12)         0.000000        -15.000000
       13)         0.000000         11.000000
       14)         0.000000          8.000000
       15)         0.000000          9.000000

 NO. ITERATIONS=      10
```

Part A-iii:

Optimal shipping routes and minimum cost can be determined by examining the results from the program:

The minimum cost is 17,100.
The shipping routes are as follows:

- P1 ships 150 units to W1
- P2 ships 200 units to W1, 250 units to W2
- P3 ships 150 units to W2 and 100 units to W3
- P4 ships 150 units to W3
- (W1 has 150+200=350 units. W2 has 250+150=400 units. W3 has 100+150=250 units)
- W1 ships 100 to R1, 150 to R2, 100 to R3. (350 units out - OK)
- W2 ships 200 to R4 and 200 R5 (400 total units out - OK)
- W3 ships 150 to R6 and 100 to R7 (250 units out OK).

Part B

We can solve this question with some simple logic. By eliminating warehouse two, we see the following simple facts:

- Warehouse 3 can receive product only from P3 and P4. That's a maximum of 400 units of supply coming in to warehouse 3 (250+150).
- The demand from retailers 5, 6, and 7 to receive from Warehouse3 is a total of 450. But there is no way for warehouse 3 to get 450 units. Those three retailers have no other warehouse if warehouse 1 is shut down.

<u>There is no *feasible solution* to this problem, therefore it is "unfeasible".</u>

We can verify this by running a modified version of the Linear programing that removes warehouse 2:

MINIMIZE $10X_{11} + 11X_{21} + 13X_{31} + 9X_{33} + 8X_{43} + 5Y_{11} + 6Y_{12} + 7Y_{13} + 10Y_{14} + 14Y_{34} + 12Y_{35} + 12Y_{36} + 6Y_{37}$

Constaints:
s.t.:

// Each Plant cannot ship more than it has
$X_{11} \leq 150$ //p1 supply
$X_{21} \leq 450$ //p2 supply
$X_{31} + X_{33} \leq 250$ //p3 supply
$X_{43} \leq 150$ //p4 supply

// Each retailer receives exactly what its demand is
$Y_{11} = 100$ // Retailer 1 demand
$Y_{12} = 150$ // Retailer 2 demand
$Y_{13} = 100$ // Retailer 3 demand
$Y_{14} + Y_{34} = 200$ // Retailer 4 demand
$Y_{35} = 200$ // Retailer 5 demand
$Y_{36} = 150$ // Retailer 6 demand
$Y_{37} = 100$ // Retailer 7 demand

// Units shipped from each warehouse must be <= amount shipped to the warehouse
$Y_{11} + Y_{12} + Y_{13} + Y_{14} - X_{11} - X_{21} - X_{31} \leq 0$
$Y_{34} + Y_{35} + Y_{36} + Y_{37} - X_{33} - X_{43} \leq 0$

Non-negativity constraints:

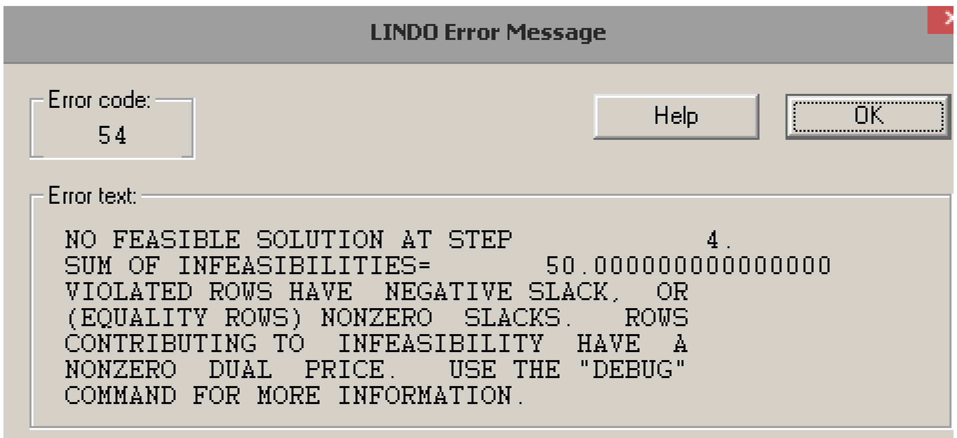
X_{pw} and $Y_{wr} \geq 0$

The LINDO code: CS325Project3Problem1b.ltx

```
MINIMIZE 10X11 + 11X21 + 13X31 + 9X33 + 8X43 + 5Y11 + 6Y12 + 7Y13 + 10Y14 + 14Y34 +
12Y35 + 12Y36 + 6Y37
ST
	X11 <= 150
	X21 <= 450
	X31 + X33 <= 250
```

```
        X43 <= 150
        Y11 = 100
        Y12 = 150
        Y13 = 100
        Y14 + Y34 = 200
        Y35 = 200
        Y36 = 150
        Y37 = 100
        Y11 + Y12 + Y13 + Y14-X11-X21-X31 <= 0
        Y34 + Y35 + Y36 + Y37-X33-X43 <= 0
```

Results:



Part C

Yes, it is possible if we limit warehouse 2 to having only 100 refrigerators per week. The optimal solution is cost 18300.00. A single constant (blue below) can be added to solve this problem. We just limit the number of units that can be shipped from each plant to warehouse 2 to be no more than 100.

MINIMIZE $10X_{11} + 15X_{12} + 11X_{21} + 8X_{22} + 13X_{31} + 8X_{32} + 9X_{33} + 14X_{42} + 8X_{43} + 5Y_{11} + 6Y_{12} + 7Y_{13} + 10Y_{14} + 12Y_{23} + 8Y_{24} + 10Y_{25} + 14Y_{26} + 14Y_{34} + 12Y_{35} + 12Y_{36} + 6Y_{37}$

Constants:

s.t.:

// New added constraint: Amount shipped to Warehouse 2 must be no greater than 100 units

$X_{12} + X_{22} + X_{32} + X_{42} \leq 100$

// Each Plant cannot ship more than it has

$X_{11} + X_{12} \leq 150$ //p1 supply

$X_{21} + X_{22} \leq 450$ //p2 supply

$X_{31} + X_{32} + X_{33} \leq 250$ //p3 supply

$X_{42} + X_{43} \leq 150$ //p4 supply

// Each retailer receives exactly what its demand is

$Y_{11} = 100$ // Retailer 1 demand

$Y_{12} = 150$ // Retailer 2 demand

$Y_{13} + Y_{23} = 100$ // Retailer 3 demand

$Y_{14} + Y_{24} + Y_{34} = 200$ // Retailer 4 demand

$Y_{25} + Y_{35} = 200$ // Retailer 5 demand

$Y_{26} + Y_{36} = 150$ // Retailer 6 demand

$Y_{37} = 100$ // Retailer 7 demand

// Units shipped from each warehouse must be <= amount shipped to the warehouse

$Y_{11} + Y_{12} + Y_{13} + Y_{14} - X_{11} - X_{21} - X_{31} \leq 0$

$Y_{23} + Y_{24} + Y_{25} + Y_{26} - X_{12} - X_{22} - X_{32} - X_{42} \leq 0$

$Y_{34} + Y_{35} + Y_{36} + Y_{37} - X_{33} - X_{43} \leq 0$

Non-negativity constraints:

X_{pw} and $Y_{wr} \geq 0$

LINDO CODE (adds one constraint to Part A code) is in CS325Project3Problem1c.ltx

```
MINIMIZE 10X11 + 15X12 + 11X21 + 8X22 + 13X31 + 8X32 + 9X33 + 14X42 + 8X43 + 5Y11 + 6Y12 +
7Y13 + 10Y14 + 12Y23 + 8Y24 + 10Y25 + 14Y26 + 14Y34 + 12Y35 + 12Y36 + 6Y37
ST
       X12 + X22 + X32 + X42 <= 100
       X11 + X12 <= 150
       X21 + X22 <= 450
       X31 + X32 + X33 <= 250
       X42 + X43 <= 150
       Y11=100
       Y12=150
       Y13 + Y23=100
       Y14 + Y24 + Y34=200
```

```
        Y25 + Y35=200
        Y26 + Y36=150
        Y37= 100
        Y11 + Y12 + Y13 + Y14-X11-X21-X31 <= 0
        Y23 + Y24 + Y25 + Y26-X12-X22-X32-X42 <= 0
        Y34 + Y35 + Y36 + Y37-X33-X43 <= 0
```

Results

```
 LP OPTIMUM FOUND AT STEP     11

        OBJECTIVE FUNCTION VALUE

        1)      18300.00

  VARIABLE        VALUE          REDUCED COST
       X11       150.000000          0.000000
       X12         0.000000          8.000000
       X21       350.000000          0.000000
       X22       100.000000          0.000000
       X31         0.000000          4.000000
       X32         0.000000          2.000000
       X33       250.000000          0.000000
       X42         0.000000          9.000000
       X43       150.000000          0.000000
       Y11       100.000000          0.000000
       Y12       150.000000          0.000000
       Y13       100.000000          0.000000
       Y14       150.000000          0.000000
       Y23         0.000000          7.000000
       Y24        50.000000          0.000000
       Y25        50.000000          0.000000
       Y26         0.000000          4.000000
       Y34         0.000000          4.000000
       Y35       150.000000          0.000000
       Y36       150.000000          0.000000
       Y37       100.000000          0.000000
```

Part D

Write out a generalized linear programming model. Give the objective function and constraints as mathematical formula.

For a given Plant-Warehouse-Retailer distribution model with n plants, m warehouses, and q retailers:

Let p be the subscript index for plants $\{P_1, P_2, \dots P_n\}$

Let w be the subscript index for Warehouses $\{W_1, W_2, \dots W_m\}$

Let r be the subscript index for Retailers $\{R_1, R_2, \dots R_q\}$

Let X_{pw} be the units to ship from P_p to W_w

Let Y_{wr} be the units to ship from W_w to R_r

Let S_p be the supply available at P_p

Let D_r be the demand at eachR_r.

Let C_{pw} be the cost of shipping from P_p to W_w.

Let K_{wr}be the cost of shipping from W_w to R_r

Objective:
Minimize the cost function, f:

$$f(n, m, q) = \sum_{w=1}^{m} \sum_{p=1}^{n} C_{pw} \cdot X_{pw} + \sum_{r=1}^{q} \sum_{w=1}^{m} K_{wr} \cdot Y_{wr}$$

Constraints:

//The sum of shipments from plant p can be no more than the supply available at plant p (S_p). Note, where there are 0 shipments from plant p to warehouse w, X_{pw} evaluates to 0.

For a given plant p, where 1 ≤ p ≤ n, this is given by the inequality:

$$\sum_{w=1}^{m} X_{pw} \leq S_p$$

//The sum of shipments to retailer r must be exactly equal to the demand at Retailer r. Note, where there are 0 shipments from warehouse w to retailer r, Y_{wr} evaluates to 0.

For a given retailer r, where 1 ≤ r ≤ q, this is given by the equality:

$$\sum_{w=1}^{m} Y_{wr} = D_r$$

//The number of units shipped from each warehouse w must be less than or equal to the amount of units shipped to the warehouse. That is, for each warehouse, the sum of outgoing shipments, Y_{wr}, minus the sum of incoming shipments, X_{pw}, must be non-positive.

For a given warehouse w, where 1 ≤ w ≤ m, this is given by the inequality:

$$\sum_{r=1}^{q} Y_{wr} - \sum_{p=1}^{n} X_{pw} \leq 0$$

//The amount shipped from any plant to any warehouse must be non-negative.

$X_{pw} \geq 0$ for all values 1 ≤ p ≤ n, and 1 ≤ w ≤ m

//The amount shipped from any warehouse to any retailer must also be non-negative.

$Y_{wr} \geq 0$ for all values 1 ≤ w ≤ m, and 1 ≤ r ≤ q.

Problem 2

Note that these variables correspond to 100 grams of a given ingredient.

That is, tm = .5 corresponds to 50 grams of tomatoes.

Tomato -	tm
Lettuce -	le
Spinach -	sp
Carrot -	ca
Sunflower Seeds -	ss
Smoked Tofu -	st
Chickpeas -	ch
Oil -	oi

At least 15 grams of protein:
****0.85*tm + 1.62*le + 2.86*sp + 0.93*ca + 23.4*ss + 16*st + 9*ch + 0*oi >=15****
-0.85*tm - 1.62*le - 2.86*sp - 0.93*ca - 23.4*ss - 16*st - 9*ch - 0*oi <= -15

At least 2 and at most 8 grams of fat:
****0.33*tm + 0.20*le + 0.39*sp + 0.24*ca + 48.7*ss + 5*st + 2.6*ch + 100*oi >= 2****
-0.33*tm - 0.20*le - 0.39*sp - 0.24*ca - 48.7*ss - 5*st - 2.6*ch - 100*oi <= -2
0.33*tm + 0.20*le + 0.39*sp + 0.24*ca + 48.7*ss + 5*st + 2.6*ch + 100*oi <= 8

At least 4 grams of carbohydrates:
****4.64*tm + 2.37*le + 3.63*sp + 9.58*ca + 15*ss + 3*st + 27*ch + 0*oi >=4****
-4.64*tm -2.37*le -3.63*sp -9.58*ca -15*ss -3*st -27*ch -0*oi <= -4

At most 200 mg of sodium:
9*tm + 28*le + 65*sp + 69*ca + 3.8*ss + 120*st + 78*ch + 0*oi <= 200

At least 40% leafy greens by mass:
****le + sp >= .4(tm + le + sp + ca + ss + st + ch + oi)****
****-.4*tm + .6*le + .6*sp -.4*ca - .4*ss -.4*st -.4*ch -.4*oi >=0****
.4*tm - .6*le - .6*sp +.4*ca + .4*ss +.4*st +.4*ch +.4*oi <=0

Cost:
f = 1*tm + .75*le + .5*sp + .5*ca + .45*ss + 2.15*st + .95*ch + 2*oi

Energy:
f = 21*tm + 16*le + 40*sp + 41*ca + 585*ss + 120*st + 164*ch + 884*oi

Part A:

Determine the combination of ingredients that minimizes calories but meets all nutritional requirements.

i. Formulate the problem as a linear program with an objective function and all constraints.

System of inequalities:
-0.85*tm - 1.62*le - 2.86*sp - 0.93*ca - 23.4*ss - 16*st - 9*ch - 0*oi <= -15
-0.33*tm - 0.20*le - 0.39*sp - 0.24*ca - 48.7*ss - 5*st - 2.6*ch - 100*oi <= -2
0.33*tm + 0.20*le + 0.39*sp + 0.24*ca + 48.7*ss + 5*st + 2.6*ch + 100*oi <= 8
-4.64*tm -2.37*le -3.63*sp -9.58*ca -15*ss -3*st -27*ch -0*oi <= -4
9*tm + 28*le + 65*sp + 69*ca + 3.8*ss + 120*st + 78*ch + 0*oi <= 200
.4*tm - .6*le - .6*sp +.4*ca + .4*ss +.4*st +.4*ch +.4*oi <=0
tm, le, sp, ca, ss, st, ch, oi >= 0 // Each of these variables may not be negative.

Objective function:

f = 21*tm + 16*le + 40*sp + 41*ca + 585*ss + 120*st + 164*ch + 884*oi

ii. Determine the optimal solution for the linear program using any software you want. Include a copy of the code/file in the report.

MATLAB: CS325Project3Problem2a.m

```
A = [-0.85 -1.62 -2.86 -0.93 -23.4 -16 -9 0
     -0.33 -0.20 -0.39 -0.24 -48.7 -5 -2.6 -100
     0.33 0.20 0.39 0.24 48.7 5 2.6 100
     -4.64 -2.37 -3.63 -9.58 -15 -3 -27 0
     9 28 65 69 3.8 120 78 0
     0.4 -0.6 -0.6 0.4 0.4 0.4 0.4 0.4];

b = [-15 -2 8 -4 200 0];

f = [21 16 40 41 585 120 164 884];

Aeq = [];
beq = [];

lb = [0, 0, 0, 0, 0, 0, 0, 0];
ub = [];

x = linprog(f,A,b,Aeq,beq,lb,ub);
```

Part A-ii Solution:

This returns x = [1.5616e-11, 0.5855, 2.7476e-11, 7.1982e-12, 1.0126e-12, 0.8782, 2.8351e-12, 4.9367e-13]

Which corresponds to 58.55 grams of Lettuce and 87.82 grams of Smoked Tofu (Appetizing?!)

This has a total of 16*.5855 + 120*.8782 = 114.752 calories, and is the minimum calorie salad that meets the nutritional requirements.

Part A-iii Solution: **What is the cost of the low calorie salad?**

<u>Cost = 0.75*.5855 + 2.15*.8782 = $2.33 per low calorie "salad".</u>

Part B**: Veronica realizes that it is also important to minimize the cost associated with the new salad. Unfortunately some of the ingredients can be expensive. Determine the combination of ingredients that minimizes cost.**

i. Formulate the problem as a linear program with an objective function and all constraints.

System of inequalities:

-0.85*tm - 1.62*le - 2.86*sp - 0.93*ca - 23.4*ss - 16*st - 9*ch - 0*oi <= -15
-0.33*tm - 0.20*le - 0.39*sp - 0.24*ca - 48.7*ss - 5*st - 2.6*ch - 100*oi <= -2
0.33*tm + 0.20*le + 0.39*sp + 0.24*ca + 48.7*ss + 5*st + 2.6*ch + 100*oi <= 8
-4.64*tm -2.37*le -3.63*sp -9.58*ca -15*ss -3*st -27*ch -0*oi <= -4
9*tm + 28*le + 65*sp + 69*ca + 3.8*ss + 120*st + 78*ch + 0*oi <= 200
.4*tm - .6*le - .6*sp +.4*ca + .4*ss +.4*st +.4*ch +.4*oi <=0
tm, le, sp, ca, ss, st, ch, oi >= 0 // Each of these variables may not be negative.

Objective function:

f = 1*tm + .75*le + .5*sp + .5*ca + .45*ss + 2.15*st + .95*ch + 2*oi

ii. Determine the optimal solution for the linear program using any software you want. Include a copy of the code/file in the report.

MATLAB: CS325Project3Problem2b.m

```
A = [-0.85 -1.62 -2.86 -0.93 -23.4 -16 -9 0
     -0.33 -0.20 -0.39 -0.24 -48.7 -5 -2.6 -100
     0.33 0.20 0.39 0.24 48.7 5 2.6 100
     -4.64 -2.37 -3.63 -9.58 -15 -3 -27 0
     9 28 65 69 3.8 120 78 0
     0.4 -0.6 -0.6 0.4 0.4 0.4 0.4 0.4];

b = [-15 -2 8 -4 200 0];

f = [1 .75 .5 .5 .45 2.15 .95 2];

Aeq = [];
beq = [];

lb = [0, 0, 0, 0, 0, 0, 0, 0];
ub = [];

x = linprog(f,A,b,Aeq,beq,lb,ub);
```

This returns x = [3.8585e-14, 1.3023e-13, 0.8323, 6.3868e-14, 0.0961, 1.2666e-13, 1.1524, 3.9520e-15]

Which corresponds to 83.23 grams of spinach, 9.61 grams of sunflower seeds, and 115.24 grams of chickpeas.

This has a total cost of .5*.8323 + .45*.0961 + .95*1.1524 = $1.56.

iii. How many calories are in the low cost salad?

Energy = 40*.8323 + 585*.0961 + 164*1.1524 = 278.5041 calories.

Part C: Compare the results from part A and B. Veronica's goal is to create a Very Veggie Salad that is both low calorie and low cost. She would like to sell the salad for $5.00 and still have a profit of at least $3.00. However if she can advertise that the salad has under 250 calories then she may be able to sell more.

i. Suggest some possible ways that she select a combination of ingredients that is "near optimal" for both objective. This is a type of multi-objective optimization.

Note that the minimum calorie salad found in part A has 114.752 calories and costs $2.33.

Note that the minimum cost salad found in part B has 278.5041 calories and costs $1.56.

Considering the constraints added in part C, we have two viable options for creating an optimized salad:

Option A: Minimize the cost for a salad that has less than 250 calories.

Modifying our code in part B by adding an additional inequality to ensure that the amount of calories is at most 250, as below:

MATLAB: **CS325Project3Problem2COptionA.m**

```
A = [-0.85 -1.62 -2.86 -0.93 -23.4 -16 -9 0
     -0.33 -0.20 -0.39 -0.24 -48.7 -5 -2.6 -100
     0.33 0.20 0.39 0.24 48.7 5 2.6 100
     -4.64 -2.37 -3.63 -9.58 -15 -3 -27 0
     9 28 65 69 3.8 120 78 0
     0.4 -0.6 -0.6 0.4 0.4 0.4 0.4 0.4
     21 16 40 41 585 120 164 884];

b = [-15 -2 8 -4 200 0 250];

f = [1 .75 .5 .5 .45 2.15 .95 2];

Aeq = [];
beq = [];

lb = [0, 0, 0, 0, 0, 0, 0, 0];
ub = [];

x = linprog(f,A,b,Aeq,beq,lb,ub);
```

We get that x = [4.2813e-14, 1.2122e-13, 0.7620, 7.3806e-14, 0.0938, 0.1689, 0.8802, 4.8830e-15]

Which corresponds to a salad with 76.20 grams of spinach, 9.38 grams of sunflower seeds, 16.89 grams of smoked tofu, and 88.02 grams of chickpeas.

This salad has 249.9738 calories, and costs $1.63

Option B: Minimize the amount of calories for a salad that costs at most $2.00 to make (so that when she sells for $5, she still makes a profit of $3)

Modifying our code in part A by adding an additional inequality to ensure that the cost is at most $2.00, as below:

MATLAB: **CS325Project3Problem2COptionB.m**

```
A = [-0.85 -1.62 -2.86 -0.93 -23.4 -16 -9 0
     -0.33 -0.20 -0.39 -0.24 -48.7 -5 -2.6 -100
     0.33 0.20 0.39 0.24 48.7 5 2.6 100
     -4.64 -2.37 -3.63 -9.58 -15 -3 -27 0
     9 28 65 69 3.8 120 78 0
     0.4 -0.6 -0.6 0.4 0.4 0.4 0.4 0.4
     1 .75 .5 .5 .45 2.15 .95 2];

b = [-15 -2 8 -4 200 0 2];

f = [21 16 40 41 585 120 164 884];

Aeq = [];
beq = [];

lb = [0, 0, 0, 0, 0, 0, 0, 0];
ub = [];

x = linprog(f,A,b,Aeq,beq,lb,ub);
```

We get that x = [4.2810e-15, 1.3774e-14, 0.5503, 5.4151e-15, 0.0294, 0.7961, 5.6316e-14, 5.6112e-16]

Which corresponds to a salad with 55.03 grams of spinach, 2.94 grams of sunflower seeds, and 79.61 grams of smoked tofu.

This salad has 134.743 calories and costs $2.00.

ii. What combination of ingredient would you suggest and what is the associated cost and calorie?

We would suggest a salad with 76.20 grams of spinach, 9.38 grams of sunflower seeds, 16.89 grams of smoked tofu, and 88.02 grams of chickpeas.

This has 249.9738 calores and costs $1.63. If sold at $5, it has a profit of $3.37 per salad sold.

iii. From a business standpoint, we can expect that her advertisement is going to have the same effect if the salad has 210 calories as it would if it had 249.99, because she is specifically advertising that her salad "has under 250 calories". Given this, it is far better to go with the salad in option A, because it is able to leverage additional profit per salad over B, even though the two salads would attract the exact same amount of business. Hence, we can safely conclude that our salad is optimal for maximizing profit with the additional constraints given in Part C.

Furthermore, this salad is almost assuredly better from a business standpoint than the minimum cost salad because of the extra boost from advertisement. The minimum cost salad (which cost $1.56, for a profit per salad of $3.44 per salad sold) only leverages an additional 7 cents per salad over the optimal profit salad found in part C. Since the profit ratio of the minimum cost salad (part B) to the optimal profit salad (part C) is 3.44/3.37 = 1.02, we can conclude that the optimal profit salad in part C will always produce more total profit than the minimum cost salad in part B, assuming that the advertising campaign described in C produces more than a 2.08% increase in salads sold.

Problem 3

Note: LINDO was used to solve all four parts of Problem 3.

a)

da = 0
db = 2
dc = 3
dd = 3
de = 9
df = 6
dg = 8
dh = 9
di = 8
dj = 10
dk = 14
dl = 15
dm = 17

Code:

```
MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm
ST
       da = 0
       db - da <= 2
       dc - da <= 3
       dd - da <= 8
       dh - da <= 9
       da - db <= 4
       dc - db <= 5
       de - db <= 7
       df - db <= 4
       dd - dc <= 10
       db - dc <= 5
       dg - dc <= 9
       di - dc <= 11
       df - dc <= 4
       da - dd <= 8
       dg - dd <= 2
       dj - dd <= 5
       df - dd <= 1
       dh - de <= 5
       dc - de <= 4
       di - de <= 10
       di - df <= 2
       dg - df <= 2
       dd - dg <= 2
       dj - dg <= 8
       dk - dg <= 12
       di - dh <= 5
       dk - dh <= 10
       da - di <= 20
       dk - di <= 6
       dj - di <= 2
       dm - di <= 12
       di - dj <= 2
```

```
        dk - dj <= 4
        dl - dj <= 5
        dh - dk <= 10
        dm - dk <= 10
        dm - dl <= 2
END
```

b) If a vertex z is added to the graph for which there is no path from vertex a to vertex z, what will be the result when you attempt to find the lengths of shortest paths as in part a).

The result will be the same, except that dz will be unbounded. This is shown in `lindo_p3b_sol.txt`. It was unnecessary and undesirable to add a constraint for dz, because since there was a path from a to all other vertices, there could be no path from any vertex to z.

Source code:

```
MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm + dz
ST
        da = 0
        db - da <= 2
        dc - da <= 3
        dd - da <= 8
        dh - da <= 9
        da - db <= 4
        dc - db <= 5
        de - db <= 7
        df - db <= 4
        dd - dc <= 10
        db - dc <= 5
        dg - dc <= 9
        di - dc <= 11
        df - dc <= 4
        da - dd <= 8
        dg - dd <= 2
        dj - dd <= 5
        df - dd <= 1
        dh - de <= 5
        dc - de <= 4
        di - de <= 10
        di - df <= 2
        dg - df <= 2
        dd - dg <= 2
        dj - dg <= 8
        dk - dg <= 12
        di - dh <= 5
        dk - dh <= 10
        da - di <= 20
        dk - di <= 6
        dj - di <= 2
        dm - di <= 12
        di - dj <= 2
        dk - dj <= 4
        dl - dj <= 5
        dh - dk <= 10
        dm - dk <= 10
```

```
        dm - dl <= 2
END
```

c) What are the lengths of the shortest paths from each vertex to vertex m. How can you solve this problem with just one linear program?

This problem is essentially a single-source shortest path problem if we simply reverse the direction of each edge but leave its weight the same. lindo_p3c.ltx shows how this could be represented as a linear program. In this case, we let dm = 0. The results were as follows, from lindo_p3c_sol.txt:

da = 17
db = 15
dc = 15
dd = 12
de = 19
df = 11
dg = 14
dh = 14
di = 9
dj = 7
dk = 10
dl = 2
dm = 0

```
MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm
ST
        dm = 0
        db - da >= -2
        dc - da >= -3
        dd - da >= -8
        dh - da >= -9
        da - db >= -4
        dc - db >= -5
        de - db >= -7
        df - db >= -4
        dd - dc >= -10
        db - dc >= -5
        dg - dc >= -9
        di - dc >= -11
        df - dc >= -4
        da - dd >= -8
        dg - dd >= -2
        dj - dd >= -5
        df - dd >= -1
        dh - de >= -5
        dc - de >= -4
        di - de >= -10
        di - df >= -2
        dg - df >= -2
        dd - dg >= -2
        dj - dg >= -8
        dk - dg >= -12
```

```
        di - dh >= -5
        dk - dh >= -10
        da - di >= -20
        dk - di >= -6
        dj - di >= -2
        dm - di >= -12
        di - dj >= -2
        dk - dj >= -4
        dl - dj >= -5
        dh - dk >= -10
        dm - dk >= -10
        dm - dl >= -2
END
```

d) Suppose that all paths must pass through vertex i. How can you calculate the length of the shortest path from any vertex x to vertex y that pass through vertex i (for all x,y V)? Calculate the lengths of these paths for the given graph. (Note for some vertices x and y it may be impossible to pass through vertex i).

This problem could be solved by finding the shortest path to vertex i from each vertex, finding the shortest path from vertex i to each vertex, and then adding each pair together. To find the shortest path through vertex i from vertex a to vertex d, for example: add the shortest path length from a to i to the shortest path length from i to d.

The results of finding the shortest path from each vertex to i are in the file lindo_p3d2_sol.txt. The results of finding the shortest path to each vertex from i are in the file lindo_p3d1_sol.txt.

Contents here of each:

```
MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm
ST
        di = 0
        db - da <= 2
        dc - da <= 3
        dd - da <= 8
        dh - da <= 9
        da - db <= 4
        dc - db <= 5
        de - db <= 7
        df - db <= 4
        dd - dc <= 10
        db - dc <= 5
        dg - dc <= 9
        di - dc <= 11
        df - dc <= 4
        da - dd <= 8
        dg - dd <= 2
        dj - dd <= 5
        df - dd <= 1
        dh - de <= 5
        dc - de <= 4
        di - de <= 10
        di - df <= 2
        dg - df <= 2
```

```
        dd - dg <= 2
        dj - dg <= 8
        dk - dg <= 12
        di - dh <= 5
        dk - dh <= 10
        da - di <= 20
        dk - di <= 6
        dj - di <= 2
        dm - di <= 12
        di - dj <= 2
        dk - dj <= 4
        dl - dj <= 5
        dh - dk <= 10
        dm - dk <= 10
        dm - dl <= 2
END
```

```
MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm
ST
        di = 0
        db - da >= -2
        dc - da >= -3
        dd - da >= -8
        dh - da >= -9
        da - db >= -4
        dc - db >= -5
        de - db >= -7
        df - db >= -4
        dd - dc >= -10
        db - dc >= -5
        dg - dc >= -9
        di - dc >= -11
        df - dc >= -4
        da - dd >= -8
        dg - dd >= -2
        dj - dd >= -5
        df - dd >= -1
        dh - de >= -5
        dc - de >= -4
        di - de >= -10
        di - df >= -2
        dg - df >= -2
        dd - dg >= -2
        dj - dg >= -8
        dk - dg >= -12
        di - dh >= -5
        dk - dh >= -10
        da - di >= -20
        dk - di >= -6
        dj - di >= -2
        dm - di >= -12
        di - dj >= -2
        dk - dj >= -4
        dl - dj >= -5
        dh - dk >= -10
        dm - dk >= -10
        dm - dl >= -2
```

```
END
```

The Python script p3d_sol.py was used to generate the table of sums that can be found in p3d_vertex_to_vertex_table.pdf.

```
import math

vertices = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm']
dist_to_i =   [8, 6, 6, 3, 10, 2, 5, 5, 0, 2, 15, math.inf, math.inf]
dist_from_i = [20, 22, 23, 28, 29, 26, 28, 16, 0, 2, 6, 7, 9]

print('\t', end = '')
for vert in vertices:
    print(str(vert) + '\t', end = '')
print()
for i in range(len(vertices)):
    print(str(vertices[i]) + '\t', end = '')
    for j in range(len(vertices)):
        print(str(dist_to_i[i] + dist_from_i[j]) + '\t', end = '')
    print()
```

	Destination Vertex													
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
Source Vertex	**a**	28	30	31	36	37	34	36	24	8	10	14	15	17
	b	26	28	29	34	35	32	34	22	6	8	12	13	15
	c	26	28	29	34	35	32	34	22	6	8	12	13	15
	d	23	25	26	31	32	29	31	19	3	5	9	10	12
	e	30	32	33	38	39	36	38	26	10	12	16	17	19
	f	22	24	25	30	31	28	30	18	2	4	8	9	11
	g	25	27	28	33	34	31	33	21	5	7	11	12	14
	h	25	27	28	33	34	31	33	21	5	7	11	12	14
	i	20	22	23	28	29	26	28	16	0	2	6	7	9
	j	22	24	25	30	31	28	30	18	2	4	8	9	11
	k	35	37	38	43	44	41	43	31	15	17	21	22	24
	l	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	m	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞