# Project 3: Linear Programming

## Problem 1: Transshipment Model

**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.

//all possible tuples of (plant, warehouse) and (warehouse, retailer) with associated costs

**Minimize:** 10(P1, W1) + 15(P1, W2) + 11(P2, W1) + 8(P2, W2) + 13(P3, W1) + 8(P3, W2) + 9(P3, W3) + 14(P4, W2) + 8(P4, W3) + 5(W1, R1) + 6(W1, R2) + 7(W1, R3) + 10(W1, R4) + 12(W2, R3) + 8(W2, R4) + 10(W2, R5) + 14(W2, R6) + 14(W3, R4) + 12(W3, R5) + 12(W3, R6) + 6(W3, R7)

**Constraints:**

//shipping capacity of each plant

(P1, W1) + (P1, W2) <= 150 //plant 1 supply

(P2, W1) + (P2, W2) <= 450 //plant 2 supply

(P3, W1) + (P3, W2) + (P3, W3) <= 250 //plant 3 supply

(P4, W2) + (P4, W3) <= 150 //plant 4 supply

//warehouses are not endpoints, and must ship all units to retailers

(P1, W1) + (P2, W1) + (P3, W1) – (W1, R1) – (W1, R2) – (W1, R3) – (W1, R4) = 0 //warehouse 1

(P1, W2) + (P2, W2) + (P3, W2) + (P4, W2) – (W2, R3) – (W2, R4) – (W2, R5) – (W2, R6) = 0 //warehouse 2

(P3, W3) + (P4, W3) – (W3, R4) – (W3, R5) – (W3, R6) – (W3, R7) = 0 //warehouse 3

//demand of retailers

(W1, R1) >= 100 //retailer 1 demand

(W1, R2) >= 150 //retailer 2 demand

(W1, R3) + (W2, R3) >= 100 //retailer 3 demand

(W1, R4) + (W2, R4) + (W3, R4) >= 200 //retailer 4 demand

(W2, R5) + (W3, R5) >= 200 //retailer 5 demand

(W2, R6) + (W3, R6) >= 150 //retailer 6 demand

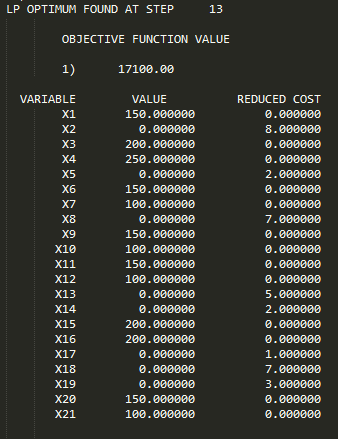
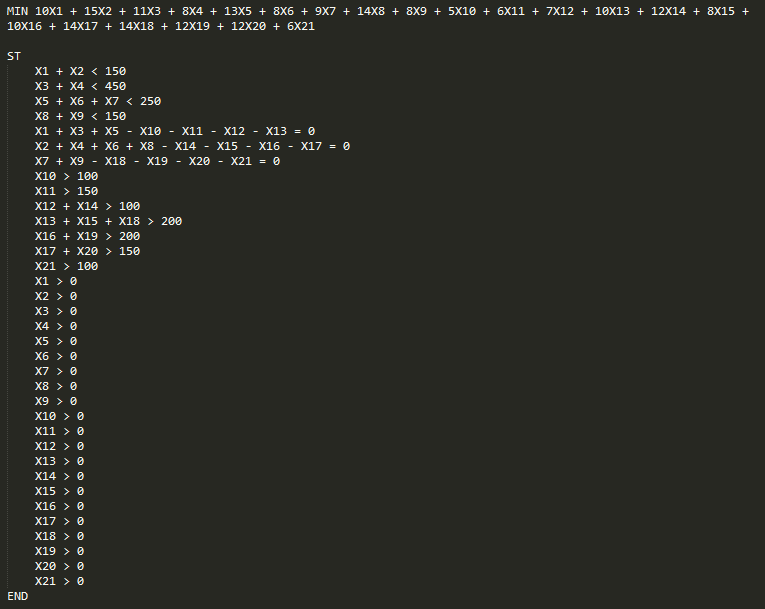
(W3, R7) >= 100 //retailer 7 demand

//nonnegativity

All tuples >= 0

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

Lindo code and results:



iii) What are the optimal shipping routes and minimum cost.

Minimum cost: $17, 100

Optimal shipping routes:

Plant 1 ships 150 units to Warehouse 1.

Plant 2 ships 200 units to Warehouse 1 and 250 units to Warehouse 2.

Plant 3 ships 150 units to Warehouse 2 and 100 units to Warehouse 3.

Plant 4 ships 150 units to Warehouse 3.

Warehouse 1 ships 100 units to Retailer 1, 150 units to Retailer 2, and 100 units to Retailer 3.

Warehouse 2 ships 200 units to Retailer 4 and 200 units to Retailer 5.

Warehouse 3 ships 150 units to Retailer 6 and 100 units to Retailer 7.

**Part B**: Due to old infrastructure Warehouse 2 is going to close eliminating all of the associated routes. What is the optimal solution for this modified model? Is it feasible to ship all the refrigerators to either warehouse 1 or 3 and then to the retailers without using warehouse 2? Why or why not?

Removing warehouse 2 from the equation results in the modified program below:

//all possible tuples of (plant, warehouse) and (warehouse, retailer) with associated costs

**Minimize:** 10(P1, W1) + 11(P2, W1) + 13(P3, W1) + 9(P3, W3) + 8(P4, W3) + 5(W1, R1) + 6(W1, R2) + 7(W1, R3) + 10(W1, R4) + 14(W3, R4) + 12(W3, R5) + 12(W3, R6) + 6(W3, R7)

**Constraints:**

//shipping capacity of each plant

(P1, W1) <= 150 //plant 1 supply

(P2, W1) <= 450 //plant 2 supply

(P3, W1) + (P3, W3) <= 250 //plant 3 supply

(P4, W3) <= 150 //plant 4 supply

//warehouses are not endpoints, and must ship all units to retailers

(P1, W1) + (P2, W1) + (P3, W1) – (W1, R1) – (W1, R2) – (W1, R3) – (W1, R4) = 0 //warehouse 1

(P3, W3) + (P4, W3) – (W3, R4) – (W3, R5) – (W3, R6) – (W3, R7) = 0 //warehouse 3

//demand of retailers

(W1, R1) >= 100 //retailer 1 demand

(W1, R2) >= 150 //retailer 2 demand

(W1, R3) >= 100 //retailer 3 demand

(W1, R4) + (W3, R4) >= 200 //retailer 4 demand

(W3, R5) >= 200 //retailer 5 demand

(W3, R6) >= 150 //retailer 6 demand

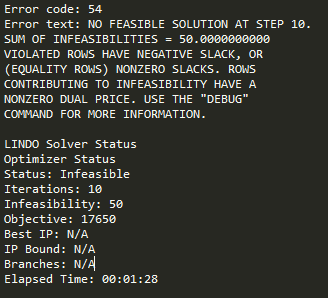
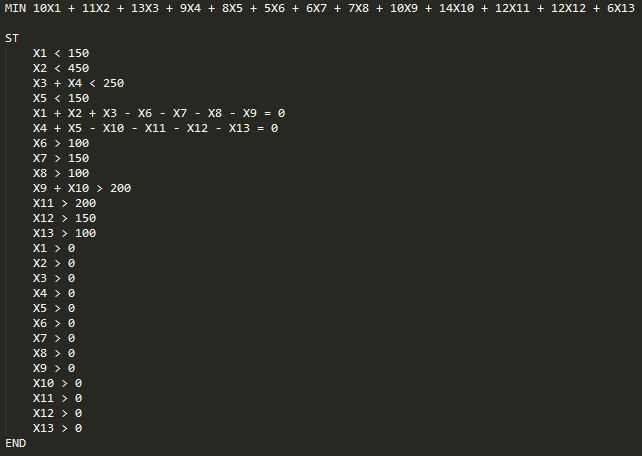
(W3, R7) >= 100 //retailer 7 demand

//nonnegativity

All tuples >= 0

It is not feasible to eliminate Warehouse 2 from the model. While all plants still have at least 1 warehouse available to ship to and all retailers are still serviced by at least 1 warehouse, Retailers 5, 6, and 7 are serviced exclusively by Warehouse 3. Even if Plan 3 and Plant 4 ship all supply to Warehouse 3, Warehouse 3 will have at most 400 units available. The combined demand from Retailers 5, 6, and 7, is 450, and so some demand (50 units) will be unmet (IE, a constraint is unsatisfiable). Therefore, there is no optimal solution.

Lindo code and error message:



**Part C**: Instead of closing Warehouse 2 management has decide to keep a portion of it open but limit shipments to 100 refrigerators per week. Is this feasible? If so what is the optimal solution when warehouse 2 is limited to 100 refrigerators?

//all possible tuples of (plant, warehouse) and (warehouse, retailer) with associated costs

**Minimize:** 10(P1, W1) + 15(P1, W2) + 11(P2, W1) + 8(P2, W2) + 13(P3, W1) + 8(P3, W2) + 9(P3, W3) + 14(P4, W2) + 8(P4, W3) + 5(W1, R1) + 6(W1, R2) + 7(W1, R3) + 10(W1, R4) + 12(W2, R3) + 8(W2, R4) + 10(W2, R5) + 14(W2, R6) + 14(W3, R4) + 12(W3, R5) + 12(W3, R6) + 6(W3, R7)

**Constraints:**

//shipping capacity of each plant

(P1, W1) + (P1, W2) <= 150 //plant 1 supply

(P2, W1) + (P2, W2) <= 450 //plant 2 supply

(P3, W1) + (P3, W2) + (P3, W3) <= 250 //plant 3 supply

(P4, W2) + (P4, W3) <= 150 //plant 4 supply

//warehouses are not endpoints, and must ship all units to retailers

(P1, W1) + (P2, W1) + (P3, W1) – (W1, R1) – (W1, R2) – (W1, R3) – (W1, R4) = 0 //warehouse 1

(P1, W2) + (P2, W2) + (P3, W2) + (P4, W2) – (W2, R3) – (W2, R4) – (W2, R5) – (W2, R6) = 0 //warehouse 2

(P3, W3) + (P4, W3) – (W3, R4) – (W3, R5) – (W3, R6) – (W3, R7) = 0 //warehouse 3

//NEW constraint – Warehouse 2 cannot receive more than 100 units

(P1, W2) + (P2, W2) + (P3, W2) + (P4, W2) <= 100

//demand of retailers

(W1, R1) >= 100 //retailer 1 demand

(W1, R2) >= 150 //retailer 2 demand

(W1, R3) + (W2, R3) >= 100 //retailer 3 demand

(W1, R4) + (W2, R4) + (W3, R4) >= 200 //retailer 4 demand

(W2, R5) + (W3, R5) >= 200 //retailer 5 demand

(W2, R6) + (W3, R6) >= 150 //retailer 6 demand

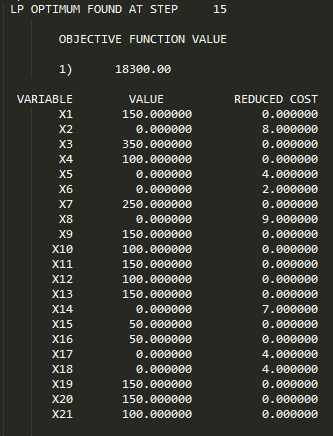
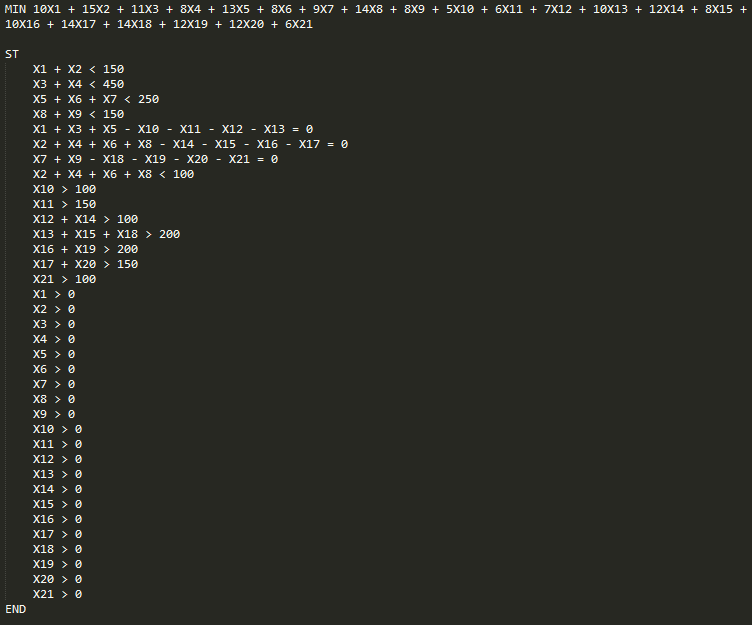
(W3, R7) >= 100 //retailer 7 demand

//nonnegativity

All tuples >= 0

Adding 100 units of capacity to Warehouse 2 solves the issue we ran into in part B, by ensuring the demands of the retailers formerly only served by Warehouse 3 can now be met.

Lindo code and report:



The optimal solution when Warehouse 2 is limited to 100 units of capacity is:

Minimum cost: $18,300

Optimal shipping routes:

Plant 1 ships 150 units to Warehouse 1.

Plant 2 ships 350 units to Warehouse 1 and 100 units to Warehouse 2.

Plant 3 ships 250 units to Warehouse 3.

Plant 4 ships 150 units to Warehouse 3.

Warehouse 1 ships 100 units to Retailer 1, 150 units to Retailer 2, 100 units to Retailer 3, and 150 units to Retailer 4.

Warehouse 2 ships 50 units to Retailer 4 and 50 units to Retailer 5.

Warehouse 3 ships 150 units to Retailer 5, 150 units to Retailer 6, and 100 units to Retailer 7.

**Part D**: Formulate a generalized linear programming model for the transshipment problem. Give the objective function and constraints as mathematical formulas.

Minimize cost(a, b) + cost(a, b+1) … for all valid values of b + cost (a+1, b) … for all valid values of a + cost(b, c) + cost(b, c+1) … for all valid values of c + cost(b+1, c) … for all valid values of b, where a = Plant #, b = Warehouse #, and c = Retailer #. A valid value is one where plant a is able to ship to warehouse b, or where warehouse b is able to ship to retailer c.

Constraints:

(a, b) + (a, b+1) … for all values of b <= capacity of a

Repeat with an additional constraint for each value of a

(a, b) + (a+1, b) … for all values of a, - (b, c) – (b, c+1) for all values of c = 0

Repeat with an additional constraint for each value of b

(b, c) + (b+1, c) … for all values of b >= demand of c

Repeat with an additional constraint for each value of c

All values of a, b, and c >= 0

## Problem 2: A Mixture Problem

*Each salad must contain:*

* *At least 15 grams of protein*
* *At least 2 and at most 8 grams of fat*
* *At least 4 grams of carbohydrates*
* *At most 200 milligrams of sodium*
* *At least 40% leafy greens by mass*

*The nutritional contents of these ingredients (per 100 grams) and cost are:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ingredient Label | Ingredient | Energy | Protein | Fat | Carbs | Sodium | Cost |
| I1 | Tomato | 21.00 | 0.85 | 0.33 | 4.64 | 9.00 | $1.00 |
| I2 | Lettuce | 16.00 | 1.62 | 0.20 | 2.37 | 28.00 | $0.75 |
| I3 | Spinach | 40.00 | 2.86 | 0.39 | 3.63 | 65.00 | $0.50 |
| I4 | Carrot | 41.00 | 0.93 | 0.24 | 9.58 | 69.00 | $0.50 |
| I5 | Sunflower Seeds | 585.00 | 23.40 | 48.70 | 15.00 | 3.80 | $0.45 |
| I6 | Smoked Tofu | 120.00 | 16.00 | 5.00 | 3.00 | 120.00 | $2.15 |
| I7 | Chickpeas | 164.00 | 9.00 | 2.60 | 27.00 | 78.00 | $0.95 |
| I8 | Oil | 884.00 | 0.00 | 100.00 | 0.00 | 0.00 | $2.00 |

*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.*

**Decision Variables**: Iy = 100 grams of each ingredient “y” to include in the salad. Each ingredient is labeled in order with the letter I and an incrementing number.

**Objective Function:** Min K = I1\*21 + I2\*16 + I3\*40 + I4\*41 + I5\*585 + I6\*120 + I7\*164 + I8\*884

Where K = kcal

**Resource Constraints:**

**Protein:** I1\*.85 + I2\*1.62 + I3\*2.86 + I4\*.93 + I5\*23.40 + I6\*16 + I7\*9 + I8\*0 ≥ 15 g of protein

**Fat Min:** I1\*.33 + I2\*.20 + I3\*.39 + I4\*.24 + I5\*48.70 + I6\*5 + I7\*2.6 + I8\*100 ≥ 2 g of fat

**Fat Max:** I1\*.33 + I2\*.20 + I3\*.39 + I4\*.24 + I5\*48.70 + I6\*5 + I7\*2.6 + I8\*100 ≤ 8 g of fat

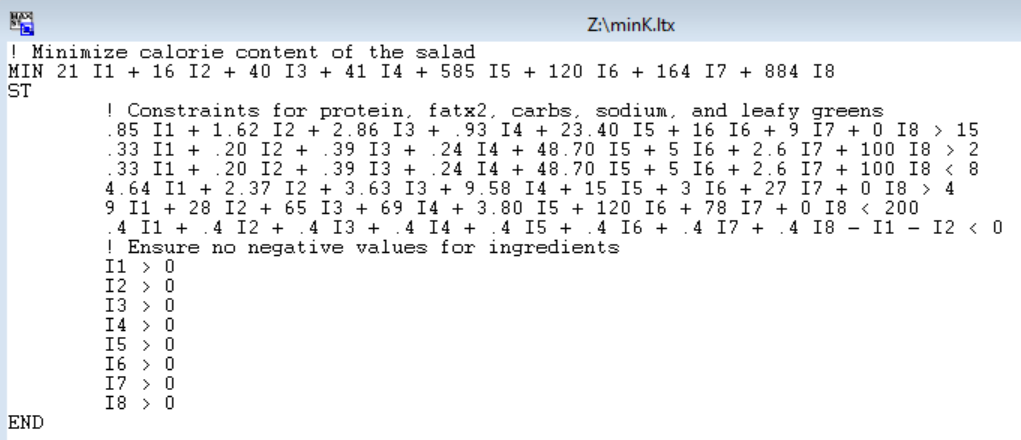
**Carbs:** I1\*4.64 + I2\*2.37 + I3\*3.63 + I4\*9.58 + I5\*15 + I6\*3 + I7\*27 + I8\*0 ≥ 4 g of carbs

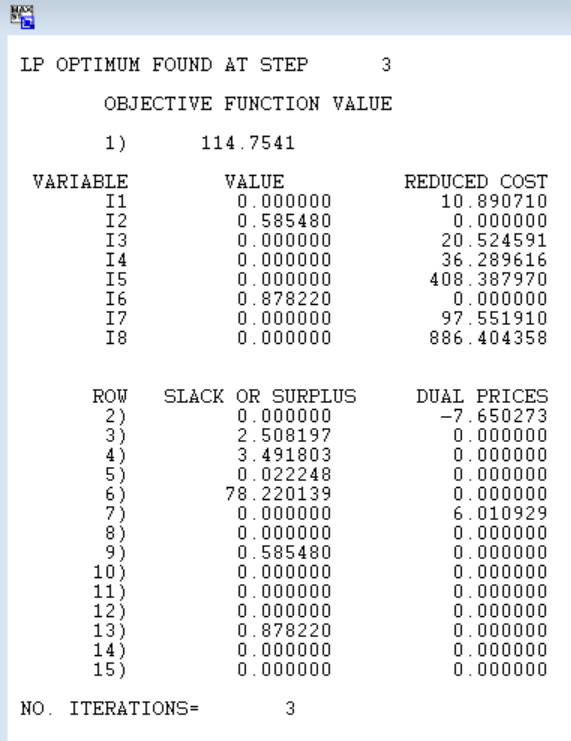
**Sodium:** I1\*9 + I2\*28 + I3\*65 + I4\*69 + I5\*3.80 + I6\*120 + I7\*78 + I8\*0 ≤ 200 mg of sodium

**Leafy Green:** (I1 + I2 + I3 + I4 + I5 + I6 + I7 + I8)\*.4 ≤ I2 + I3

**Non-Negative:** Iy ≥ 0

*ii) Screenshots of code used to determine the optimal solution*

**

**

*iii) What is the cost of the low calorie salad?*

The solution is 58.55 grams of Lettuce @ $0.75/100g and 87.82 grams of Smoked Tofu @ $2.15/100g. This results in calories of 114.75 kcal for a **total cost of $2.33.**

*Part B: 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.*

**Decision Variables**: Iy = 100 grams of each ingredient “y” to include in the salad. Each ingredient is labeled in order with the letter I and an incrementing number.

**Objective Function:** Min D = I1\*1.00 + I2\*.75 + I3\*.50 + I4\*.50 + I5\*.45 + I6\*2.15 + I7\*.95 + I8\*2.00

Where D = dollars spent

**Resource Constraints:**

**Protein:** I1\*.85 + I2\*1.62 + I3\*2.86 + I4\*.93 + I5\*23.40 + I6\*16 + I7\*9 + I8\*0 ≥ 15 g of protein

**Fat Min:** I1\*.33 + I2\*.20 + I3\*.39 + I4\*.24 + I5\*48.70 + I6\*5 + I7\*2.6 + I8\*100 ≥ 2 g of fat

**Fat Max:** I1\*.33 + I2\*.20 + I3\*.39 + I4\*.24 + I5\*48.70 + I6\*5 + I7\*2.6 + I8\*100 ≤ 8 g of fat

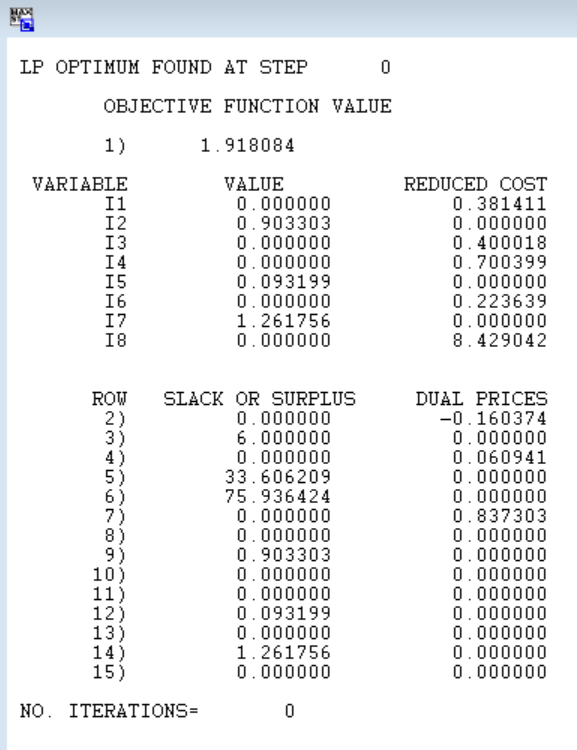
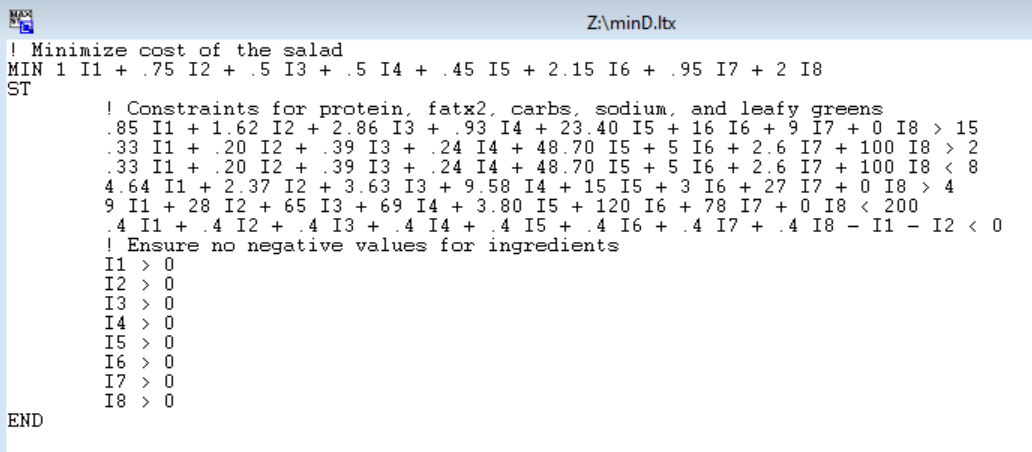
**Carbs:** I1\*4.64 + I2\*2.37 + I3\*3.63 + I4\*9.58 + I5\*15 + I6\*3 + I7\*27 + I8\*0 ≥ 4 g of carbs

**Sodium:** I1\*9 + I2\*28 + I3\*65 + I4\*69 + I5\*3.80 + I6\*120 + I7\*78 + I8\*0 ≤ 200 mg of sodium

**Leafy Green:** (I1 + I2 + I3 + I4 + I5 + I6 + I7 + I8)\*.4 ≤ I2 + I3

**Non-Negative:** Iy ≥ 0

*ii) Screenshots of code used to determine the optimal solution*

**

*iii) How many calories are in the low cost salad?*

The solution is 90.33 grams of Lettuce @ 16 kcal/100g, 9.32 grams of Sunflower Seeds @ 585 kcal/100g, and 126.18 grams of Chickpeas @ 164 kcal/100g. This results in a total cost of $1.92 and **275.91 kcal for the salad.**

*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.*

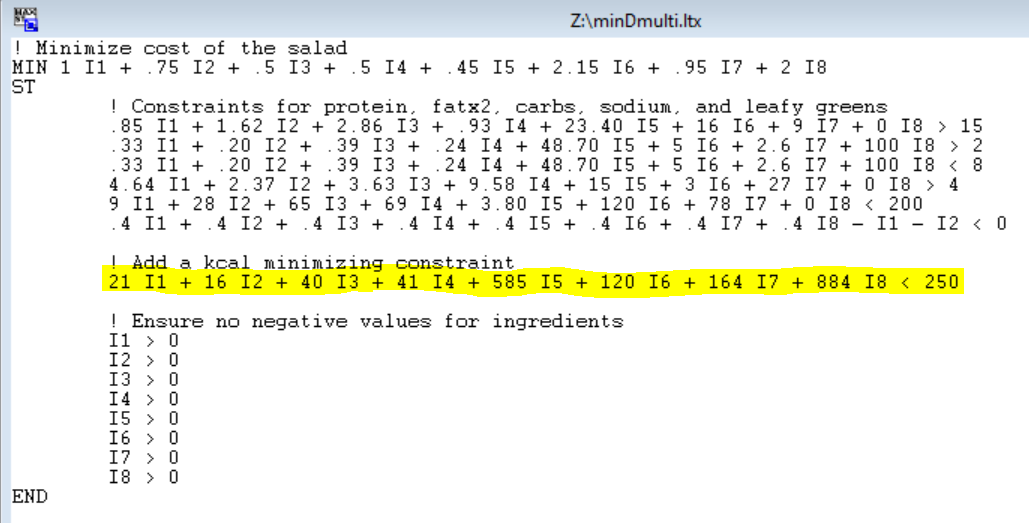
|  |  |  |
| --- | --- | --- |
|  | Low Calorie | Low Cost |
| Kcal | 114.75 | 275.91 |
| Total Cost | $2.33 | $1.92 |

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

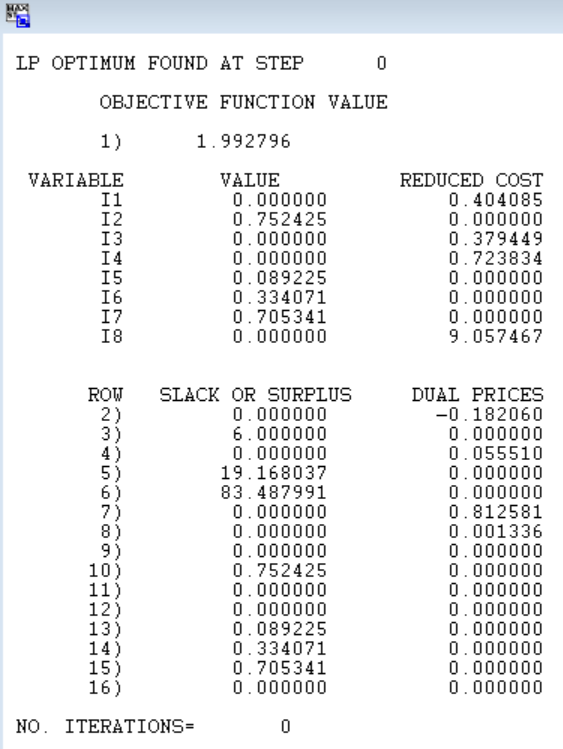
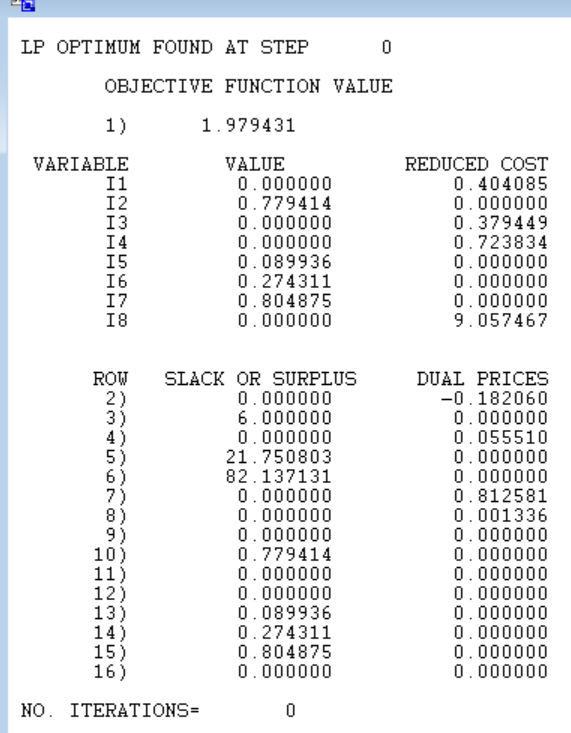
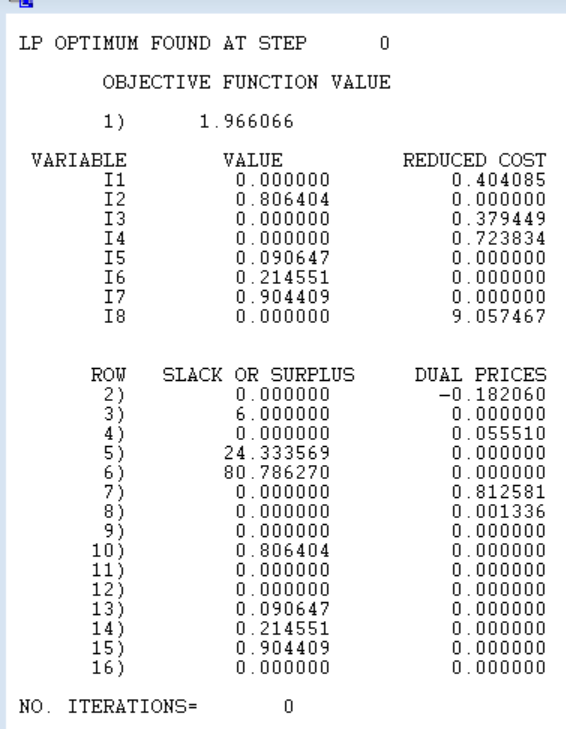
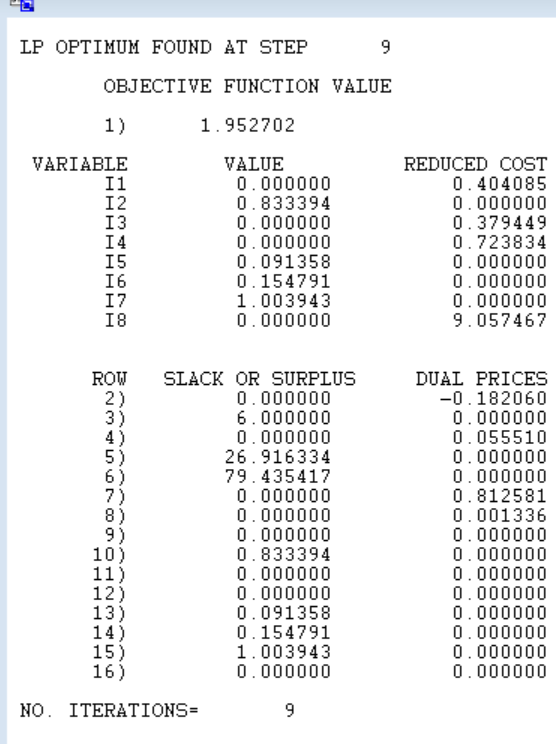
To create a Linear Programming problem that can help the user solve for both of these items, the objective from on problem should become a constraint in the other problem. Typically there might be something specific that is driving the user to decide which one should be the constraints. In this case, the 2 goals are a salad that costs < $2.00, and a salad that has under 250 kcal. In this case, the Low Calorie option is well below the < 250 calorie goal and exceeds the $2.00 cost benchmark. Meanwhile, the Low Cost option is close to the kcal goal at 275.91 and costs $1.92, just barely under the cost goal. Since the low cost solution is near optimal, I would recommend the user adds the low calorie constraint to the low cost problem. The user can then manually modify the low calorie constraint to fine tune the desired results. The user can continue to tighten (improve) the low calorie constraint until the increase in total cost is undesirable (the user will have to be able to decide which is more important after a certain point).

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

The problem setup:



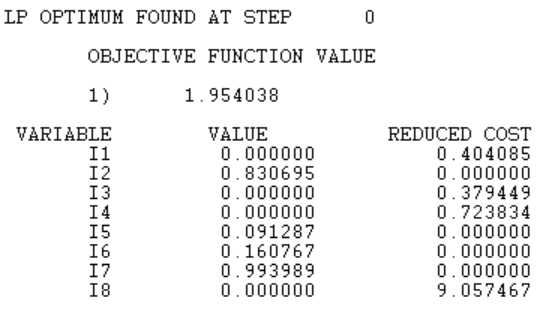
Examples of Solutions:



Results Table:

|  |  |  |
| --- | --- | --- |
| Calorie Constraint | Total kcal | Total Cost |
| <250 | 250.00 | $1.95 |
| <240 | 240.00 | $1.97 |
| <230 | 230.00 | $1.98 |
| <220 | 220.00 | $1.99 |

From a business perspective, assuming no incremental gain from lowering calories below 249 (one below 250 so the business is not caught lying and slandered in the media), the optimal solution would be 249 kcal at a cost of $1.95. This is achieved by using 83.07 grams of Lettuce, 9.13 grams of Sunflower Seeds, 16.08 grams of Smoked Tofu, and 99.40 grams of Chickpeas.



*iii) Note: There is not one “right” answer. Discuss how you derived your solution.*

As noted above, the solution was derived through a series of guess and check activities, starting with the minimum accepted answer (kcal below 250 to increase sales). From there, it was apparent that lowering kcal would result in increased costs. Since there is no incremental gain listed between 250 kcal and 220 kcal for this problem, than it is not worth incurring the extra cost. In a real world scenario, it might be worth using the lower kcal values at higher cost because the added marketing leverage could potentially increase sales.

## Problem 3: Solving Shortest Path Problems Using Linear Programming

1. *What are the lengths of the shortest paths from vertex a to all other vertices.*

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

da > 0

db > 0

dc > 0

dd > 0

de > 0

df > 0

dg > 0

dh > 0

di > 0

dj > 0

dk > 0

dl > 0

dm > 0

END

VARIABLE VALUE REDUCED COST

DA 0.000000 0.000000

DB 2.000000 0.000000

DC 3.000000 0.000000

DD 8.000000 0.000000

DE 9.000000 0.000000

DF 6.000000 0.000000

DG 8.000000 0.000000

DH 9.000000 0.000000

DI 8.000000 0.000000

DJ 10.000000 0.000000

DK 14.000000 0.000000

DL 15.000000 0.000000

DM 17.000000 0.000000

1. *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 distance value dz for that vertex is unbounded because there are no constraints imposed on its value by edge weights, so you can increase the maximum forever by just increasing dz.

1. *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?*

MAX da + db + dc + dd + de + df + dg + dh + di + dj + dk + dl + dm

ST

dm = 0

da - db < 2

da - dc < 3

da - dd < 8

da - dh < 9

db - da < 4

db - dc < 5

db - de < 7

db - df < 4

dc - dd < 10

dc - db < 5

dc - dg < 9

dc - di < 11

dc - df < 4

dd - da < 8

dd - dg < 2

dd - dj < 5

dd - df < 1

de - dh < 5

de - dc < 4

de - di < 10

df - di < 2

df - dg < 2

dg - dd < 2

dg - dj < 8

dg - dk < 12

dh - di < 5

dh - dk < 10

di - da < 20

di - dk < 6

di - dj < 2

di - dm < 12

dj - di < 2

dj - dk < 4

dj - dl < 5

dk - dh < 10

dk - dm < 10

dl - dm < 2

da > 0

db > 0

dc > 0

dd > 0

de > 0

df > 0

dg > 0

dh > 0

di > 0

dj > 0

dk > 0

dl > 0

END

VARIABLE VALUE REDUCED COST

DA 17.000000 0.000000

DB 15.000000 0.000000

DC 15.000000 0.000000

DD 12.000000 0.000000

DE 19.000000 0.000000

DF 11.000000 0.000000

DG 14.000000 0.000000

DH 14.000000 0.000000

DI 9.000000 0.000000

DJ 7.000000 0.000000

DK 10.000000 0.000000

DL 2.000000 0.000000

DM 0.000000 0.000000

For this version, instead of starting with a source vertex s and finding the shortest paths to all other vertices in the graph, we start from target vertex t and find all vertices that point to t, and work our way outward from there. The LINDO program is almost identical, except the target is set with the “= 0” constraint and we swap the operands of the subtraction operator in all of the edge constraints to reverse the direction.

1. *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).*

We can use the same linear program as part c) and set the target to vertex i. This gives us the shortest path from every reachable vertex u to vertex i. Any unreachable vertex is unbounded. Then we run the same linear program as part a) and set the source to vertex i. This gives us the shortest path from vertex i to every reachable vertex v. Once again, any unreachable vertex is unbounded. Finally, we simply add the values together for every permutation (u, i, v) for every u, v ∈ V to get the shortest path from u to v via i.

VARIABLE SP TO DI SP FROM DI

DA 8.000000 20.000000

DB 6.000000 22.000000

DC 6.000000 23.000000

DD 3.000000 28.000000

DE 10.000000 29.000000

DF 2.000000 26.000000

DG 5.000000 28.000000

DH 5.000000 16.000000

DI 0.000000 0.000000

DJ 2.000000 2.000000

DK 15.000000 6.000000

DL UNBOUNDED 7.000000

DM UNBOUNDED 9.000000

Shortest Path Via i (δ(u,i,v))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v  u | a | b | c | d | e | f | g | h | i | j | k | l | m |
| 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 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| m | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |