Chris Merrill

CS325

11/8/2017

**Homework Assignment 6**

**NOTE: ALL CODE IS INCLUDED AT THE BOTTOM OF THIS FILE**

**1. Shortest Paths using LP: *(7 points)* Shortest paths can be cast as an LP using distances dv from the source s to a particular vertex v as variables.**

* **We can compute the shortest path from s to t in a weighted directed graph by solving.**

**max dt**

**subject to**

**ds = 0**

**dv – du ≤ w(u,v) for all (u,v)E**

* **We can compute the single-source by changing the objective function to**

**max ∑𝑣∈𝑉 𝑑𝑣**

**Use linear programming to answer the questions below. Submit a copy of the LP code and output.**

1. **Find the distance of the shortest path from G to C in the graph below.**

To compute the length of the shortest path from G to C in a weighted directed graph we can use:

Maximize dc

Subject To dg = 0

dv – du ≤ lu→v for every edge u → v

I used LINDO to carry out the calculation (see attached code and output) and found that:

Shortest Path from G to C = 16

1. **Find the distances of the shortest paths from G to all other vertices.**

Using the same method as in a, the shortest paths to all remaining points are:

G → A = 7

G → B = 12

G → C = 16

G → D = 2

G → E = 19

G → F = 17

G → H = 3



**25**



**A**



**H**



**B**



**F**



**E**



**D**



**C**



**G**



**9**



**7**



**2**



**10**



**18**



**3**



**4**



**3**



**7**



**5**



**8**



**9**



**4**



**3**



**10**



**2**

1. **Product Mix: (*7 points)* Acme Industries produces four types of men’s ties using three types of material. Your job is to determine how many of each type of tie to make each month. The goal is to maximize profit, profit per tie = selling price - labor cost – material cost. Labor cost is $0.75 per tie for all four types of ties. The material requirements and costs are given below.**

|  |  |  |
| --- | --- | --- |
| **Material** | **Cost per yard** | **Yards available per month** |
| Silk | $20 | 1,000 |
| Polyester | $6 | 2,000 |
| Cotton | $9 | 1,250 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Information** | **Type of Tie** | |  |  |
| **Silk = s** | **Poly = p** | **Blend1 = b** | **Blend2 = c** |
| Selling Price per tie | $6.70 | $3.55 | $4.31 | $4.81 |
| Monthly Minimum units | 6,000 | 10,000 | 13,000 | 6,000 |
| Monthly Maximum units | 7,000 | 14,000 | 16,000 | 8,500 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Material**  **Information in yards** |  | **Type of Tie** | |  |
| **Silk** | **Polyester** | **Blend 1 (50/50)** | **Blend 2 (30/70)** |
| Silk | 0.125 | 0 | 0 | 0 |
| Polyester | 0 | 0.08 | 0.05 | 0.03 |
| Cotton | 0 | 0 | 0.05 | 0.07 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| type | selling price | labor | material | profit per tie |
| silk s | 6.7 | 0.75 | 2.5 | 3.45 |
| poly p | 3.55 | 0.75 | 0.48 | 2.32 |
| blend1 b | 4.31 | 0.75 | 0.75 | 2.81 |
| blend2 c | 4.81 | 0.75 | 0.81 | 3.25 |

**Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. What are the optimal numbers of ties of each type to maximize profit? Include a copy of the code and output.**

I used LINDO to solve the following (see attached code and output):

|  |  |  |
| --- | --- | --- |
| Maximize | 3.45s + 2.32p + 2.81b + 3.25c |  |
| Subject To | 0.125s ≤ 1000 | (1000 yards of Silk available each month) |
|  | 0.08s + 0.05b + 0.03c ≤ 2000 | (2000 yards of Polyester available each month) |
|  | 0.05b + 0.07c ≤ 1250 | (1250 yards of Cotton available each month) |
|  | s ≤ 7000 | (Silk has a monthly max of 7000 units) |
|  | s ≥ 6000 | (Silk has a monthly min of 6000 units) |
|  | p ≤ 14000 | (Polyester has a monthly max of 14000 units) |
|  | p ≥ 10000 | (Polyester has a monthly min of 10000 units) |
|  | b ≤ 16000 | (Cotton blend 1 has a monthly max of 16000 units) |
|  | b ≥ 13000 | (Cotton blend 1 has a monthly min of 13000 units) |
|  | c ≤ 8500 | (Cotton Blend 2 has a monthly max of 8500 units) |
|  | c ≥ 6000 | (Cotton Blend 2 has a monthly min of 6000 units) |

Solution (Optimal # of Ties to Maximize Profit):

|  |  |  |
| --- | --- | --- |
| **Type** | **# to Make** | **Value** |
| Silk | 7000 | $24,150 |
| Polyester | 13625 | $31,610 |
| Blend 1 | 13100 | $36,811 |
| Blend 2 | 8500 | $27,625 |
|  | **TOTAL:** | **$120,196** |

**3. Transshipment Model *(7 points)***

**This is an extension of the transportation model. There are now intermediate transshipment points added between the sources (plants) and destinations (retailers). Items being shipped from a Plant (*pi*) must be shipped to a Warehouse (*wj*) before being shipped to the Retailer (*rk*). Each Plant will have an associated supply (*si*) and each Retailer will have a demand (*dk*). The number of plants is n, number of warehouses is q and the number of retailers is m. The edges (*i,j*) from plant (*pi*)to warehouse (*wj*) have costs associated denoted cp(*i,j*). The edges (*j,k*) from a warehouse (*wj*)to a retailer (*rk*) have costs associated denoted cw(*j,k*).**

**The graph below shows the transshipment map for a manufacturer of refrigerators. Refrigerators are produced at four plants and then shipped to a warehouse (weekly) before going to the retailer.**



P1



P2



W3



W2



P4



P3



W1



R1



R2



R3



R4



R6



R5



R7

**Below are the costs of shipping from a plant to a warehouse and then a warehouse to a retailer. If it is impossible to ship between the two locations an X is placed in the table.**

|  |  |  |  |
| --- | --- | --- | --- |
| **cost** | **W1** | **W2** | **W3** |
| **P1** | $10 | $15 | X |
| **P2** | $11 | $8 | X |
| **P3** | $13 | $8 | $9 |
| **P4** | X | $14 | $8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **cost** | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** | **R7** |
| **W1** | $5 | $6 | $7 | $10 | X | X | X |
| **W2** | X | X | $12 | $8 | $10 | $14 | X |
| **W3** | X | X | X | $14 | $12 | $12 | $6 |

**The tables below give the capacity of each plant (supply) and the demand for each retailer (per week).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** |
| **Supply** | 150 | 450 | 250 | 150 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** | **R7** |
| **Demand** | 100 | 150 | 100 | 200 | 200 | 150 | 100 |

**Your goal is to determine the number of refrigerators to be shipped plants to warehouses and then warehouses to retailers to minimize the cost. Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. What are the optimal shipping routes and minimum cost.**

**Include a copy of the code and output.**

I used LINDO to solve the following (see attached code and output):

Minimize 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 +

8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 +

14W2R6 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7

|  |  |
| --- | --- |
| Subject To | P1W1 + P2W1 + P3W1 - W1R1 - W1R2 - W1R3 - W1R4 ≥ 0 |
|  | P1W2 + P2W2 + P3W2 + P4W2 - W2R3 - W2R4 - W2R5 - W2R6 ≥ 0 |
|  | P3W3 + P4W3 - W3R4 - W3R5 - W3R6 - W3R7 ≥ 0 |
|  | P1W1 + P1W2 ≤ 150 |
|  | P2W1 + P2W2 ≤ 450 |
|  | P3W1 + P3W2 + P3W3 ≤ 250 |
|  | P4W2 + P4W3 ≤ 150 |
|  | W1R1 ≥ 100 |
|  | W1R2 ≥ 150 |
|  | W1R3 + W2R3 ≥ 100 |
|  | W1R4 + W2R4 + W3R4 ≥ 200 |
|  | W2R5 + W3R5 ≥ 200 |
|  | W2R6 + W3R6 ≥ 150 |
|  | W3R7 ≥ 100 |

The first 3 constraints assure that the refrigerators shipped from each plant are ≥ 0. The next three constraints assure that the refrigerators shipped from each plant are less than or equal to the available supply. The final seven constraints assure that the demands are met.

Optimal Shipping Routes:

150 from P1 → W1

200 from P2 → W1

250 from P2 → W2

150 from P3 → W2

100 from P3 → W3

150 from P4 → W3

100 from W1 → R1

150 from W1 → R2

100 from W1 → R3

200 from W2 → R4

200 from W2 → R5

150 from W3 → R6

100 from W3 → R7

Total Minimum Cost: $17,100.00

**4: A Mixture Problem *(9 points)***

**Veronica the owner of Very Veggie Vegeria is creating a new healthy salad that is low in calories but meets certain nutritional requirements. A salad is any combination of the following ingredients: Tomato, Lettuce, Spinach, Carrot, Smoked Tofu, Sunflower Seeds, Chickpeas, Oil**

**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** | **Energy (Cal)** | **Protein**  **(grams)** | **Fat (grams)** | **Carbohydrate (grams)** | **Sodium (mg)** | **Cost (100g)** |
| **Tomato** | 21 | 0.85 | 0.33 | 4.64 | 9.00 | $1.00 |
| **Lettuce** | 16 | 1.62 | 0.20 | 2.37 | 28.00 | $0.75 |
| **Spinach** | 40 | 2.86 | 0.39 | 3.63 | 65.00 | $0.50 |
| **Carrot** | 41 | 0.93 | 0.24 | 9.58 | 69.00 | $0.50 |
| **Sunflower Seeds** | 585 | 23.4 | 48.7 | 15.00 | 3.80 | $0.45 |
| **Smoked Tofu** | 120 | 16.00 | 5.00 | 3.00 | 120.00 | $2.15 |
| **Chickpeas** | 164 | 9.00 | 2.6 | 27.0 | 78.00 | $0.95 |
| **Oil** | 884 | 0 | 100.00 | 0 | 0 | $2.00 |

**Part A**: **Determine the combination of ingredients that minimizes calories but meets all nutritional requirements. Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. What is the cost of the low-calorie salad?**

I used LINDO to solve the following (see attached code and output):

Minimize 21T + 16L + 40S + 41C + 585SS + 120ST + 164CP + 884O

|  |  |  |
| --- | --- | --- |
| Subject To | 0.85T + 1.62L + 2.86S + 0.93C + 23.4SS + 16ST + 9CP ≥ 15 | (at least 15g of protein) |
|  | 0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O ≥ 2 | (at least 2g of fat) |
|  | 0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O ≤ 8 | (at most 8g of fat) |
|  | 4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP >= 4 | (at least 4g of carbs) |
|  | 9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP ≤ 200 | (at most 200mg of sodium) |
|  | 0.6L + 0.6S - 0.4T - 0.4C - 0.4SS - 0.4ST -0.4CP - 0.40O >= 0 | (40% leafy greens by mass) |

Optimal Solution:

58.584g of Lettuce

87.822g of Smoked Tofu

Total Calories: 114.7541

Total Cost: $2.33

**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. Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. How many calories are in the low-cost salad?**

I used LINDO to solve the following (see attached code and output):

Minimize 1T + 0.75L + 0.5S + 0.5C + 0.45SS + 2.15ST + 0.95CP + 2O

|  |  |  |
| --- | --- | --- |
| Subject To | 0.85T + 1.62L + 2.86S + 0.93C + 23.4SS + 16ST + 9CP ≥ 15 | (at least 15g of protein) |
|  | 0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O ≥ 2 | (at least 2g of fat) |
|  | 0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O ≤ 8 | (at most 8g of fat) |
|  | 4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP >= 4 | (at least 4g of carbs) |
|  | 9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP ≤ 200 | (at most 200mg of sodium) |
|  | 0.6L + 0.6S - 0.4T - 0.4C - 0.4SS - 0.4ST -0.4CP - 0.40O >= 0 | (40% leafy greens by mass) |

Optimal Solution:

83.2298g of Spinach

9.6083g of Sunflower Seeds

115.2364g of Chick Peas

Total Cost: $1.55

Total Calories: 278.488171

**LINDO Code and Output for each problem:**

**Problem 1 Part A Code**

max dc

st

dg = 0

dg - de <= 7

dd - dg <= 2

dh - dg <= 3

da - dh <= 4

db - dh <= 9

de - db <= 10

db - da <= 8

dd - de <= 9

de - dd <= 25

dd - dc <= 3

df - dd <= 18

dc - db <= 4

dc - df <= 3

db - df <= 7

df - da <= 10

da - df <= 5

de - df <= 2

end

**Problem 1 Part A Output**

LP OPTIMUM FOUND AT STEP 0

OBJECTIVE FUNCTION VALUE

1) 16.00000

VARIABLE VALUE REDUCED COST

DC 16.000000 0.000000

DG 0.000000 0.000000

DE 0.000000 0.000000

DD 0.000000 0.000000

DH 3.000000 0.000000

DA 4.000000 0.000000

DB 12.000000 0.000000

DF 13.000000 0.000000

ROW SLACK OR SURPLUS DUAL PRICES

2) 0.000000 1.000000

3) 7.000000 0.000000

4) 2.000000 0.000000

5) 0.000000 1.000000

6) 3.000000 0.000000

7) 0.000000 1.000000

8) 22.000000 0.000000

9) 0.000000 0.000000

10) 9.000000 0.000000

11) 25.000000 0.000000

12) 19.000000 0.000000

13) 5.000000 0.000000

14) 0.000000 1.000000

15) 0.000000 0.000000

16) 8.000000 0.000000

17) 1.000000 0.000000

18) 14.000000 0.000000

19) 15.000000 0.000000

NO. ITERATIONS= 0

**Problem 1 Part B Code**

max da + db + dc + dd + de + df + dh

st

dg = 0

dg - de <= 7

dd - dg <= 2

dh - dg <= 3

da - dh <= 4

db - dh <= 9

de - db <= 10

db - da <= 8

dd - de <= 9

de - dd <= 25

dd - dc <= 3

df - dd <= 18

dc - db <= 4

dc - df <= 3

db - df <= 7

df - da <= 10

da - df <= 5

de - df <= 2

end

**Problem 1 Part B Output**

LP OPTIMUM FOUND AT STEP 2

OBJECTIVE FUNCTION VALUE

1) 76.00000

VARIABLE VALUE REDUCED COST

DA 7.000000 0.000000

DB 12.000000 0.000000

DC 16.000000 0.000000

DD 2.000000 0.000000

DE 19.000000 0.000000

DF 17.000000 0.000000

DH 3.000000 0.000000

DG 0.000000 0.000000

ROW SLACK OR SURPLUS DUAL PRICES

2) 0.000000 7.000000

3) 26.000000 0.000000

4) 0.000000 1.000000

5) 0.000000 6.000000

6) 0.000000 3.000000

7) 0.000000 2.000000

8) 3.000000 0.000000

9) 3.000000 0.000000

10) 26.000000 0.000000

11) 8.000000 0.000000

12) 17.000000 0.000000

13) 3.000000 0.000000

14) 0.000000 1.000000

15) 4.000000 0.000000

16) 12.000000 0.000000

17) 0.000000 2.000000

18) 15.000000 0.000000

19) 0.000000 1.000000

NO. ITERATIONS= 2

**Problem 2 Code**

max 3.45s + 2.32p + 2.81b + 3.25c

st

0.125s <= 1000

0.08p + 0.05b + 0.03c <= 2000

0.05b + 0.07c <= 1250

s <= 7000

s >= 6000

p <= 14000

p >= 10000

b <= 16000

b >= 13000

c <= 8500

c >= 6000

end

**Problem 2 Output**

LP OPTIMUM FOUND AT STEP 2

OBJECTIVE FUNCTION VALUE

1) 120196.0

VARIABLE VALUE REDUCED COST

S 7000.000000 0.000000

P 13625.000000 0.000000

B 13100.000000 0.000000

C 8500.000000 0.000000

ROW SLACK OR SURPLUS DUAL PRICES

2) 125.000000 0.000000

3) 0.000000 29.000000

4) 0.000000 27.200001

5) 0.000000 3.450000

6) 1000.000000 0.000000

7) 375.000000 0.000000

8) 3625.000000 0.000000

9) 2900.000000 0.000000

10) 100.000000 0.000000

11) 0.000000 0.476000

12) 2500.000000 0.000000

NO. ITERATIONS= 2

**Problem 3 Code**

min 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7

st

P1W1 + P2W1 + P3W1 - W1R1 - W1R2 - W1R3 - W1R4 >= 0

P1W2 + P2W2 + P3W2 + P4W2 - W2R3 - W2R4 - W2R5 - W2R6 >= 0

P3W3 + P4W3 - W3R4 - W3R5 - W3R6 - W3R7 >= 0

P1W1 + P1W2 <= 150

P2W1 + P2W2 <= 450

P3W1 + P3W2 + P3W3 <= 250

P4W2 + P4W3 <= 150

W1R1 >= 100

W1R2 >= 150

W1R3 + W2R3 >= 100

W1R4 + W2R4 + W3R4 >= 200

W2R5 + W3R5 >= 200

W2R6 + W3R6 >= 150

W3R7 >= 100

end

**Problem 3 Output**

LP OPTIMUM FOUND AT STEP 13

OBJECTIVE FUNCTION VALUE

1) 17100.00

VARIABLE VALUE REDUCED COST

P1W1 150.000000 0.000000

P1W2 0.000000 8.000000

P2W1 200.000000 0.000000

P2W2 250.000000 0.000000

P3W1 0.000000 2.000000

P3W2 150.000000 0.000000

P3W3 100.000000 0.000000

P4W2 0.000000 7.000000

P4W3 150.000000 0.000000

W1R1 100.000000 0.000000

W1R2 150.000000 0.000000

W1R3 100.000000 0.000000

W1R4 0.000000 5.000000

W2R3 0.000000 2.000000

W2R4 200.000000 0.000000

W2R5 200.000000 0.000000

W2R6 0.000000 1.000000

W3R4 0.000000 7.000000

W3R5 0.000000 3.000000

W3R6 150.000000 0.000000

W3R7 100.000000 0.000000

ROW SLACK OR SURPLUS DUAL PRICES

2) 0.000000 -11.000000

3) 0.000000 -8.000000

4) 0.000000 -9.000000

5) 0.000000 1.000000

6) 0.000000 0.000000

7) 0.000000 0.000000

8) 0.000000 1.000000

9) 0.000000 -16.000000

10) 0.000000 -17.000000

11) 0.000000 -18.000000

12) 0.000000 -16.000000

13) 0.000000 -18.000000

14) 0.000000 -21.000000

15) 0.000000 -15.000000

NO. ITERATIONS= 13

**Problem 4 Part A Code**

min 21T + 16L + 40S + 41C + 585SS + 120ST + 164CP + 884O

st

0.85T + 1.62L + 2.86S + 0.93C + 23.4SS + 16ST + 9CP >= 15

0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O >= 2

0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O <= 8

4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP >= 4

9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP <= 200

0.6L + 0.6S - 0.4T - 0.4C - 0.4SS - 0.4ST -0.4CP - 0.40O >= 0

End

**Problem 4 Part A Output**

LP OPTIMUM FOUND AT STEP 5

OBJECTIVE FUNCTION VALUE

1) 114.7541

VARIABLE VALUE REDUCED COST

T 0.000000 16.901640

L 0.585480 0.000000

S 0.000000 14.513662

C 0.000000 36.289616

SS 0.000000 408.387970

ST 0.878220 0.000000

CP 0.000000 97.551910

O 0.000000 886.404358

ROW SLACK OR SURPLUS DUAL PRICES

2) 0.000000 -7.650273

3) 2.508197 0.000000

4) 3.491803 0.000000

5) 0.022248 0.000000

6) 78.220139 0.000000

7) 0.000000 -6.010929

NO. ITERATIONS= 5

**Problem 4 Part B Code**

min 1T + 0.75L + 0.5S + 0.5C + 0.45SS + 2.15ST + 0.95CP + 2O

st

0.85T + 1.62L + 2.86S + 0.93C + 23.4SS + 16ST + 9CP >= 15

0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O >= 2

0.33T + 0.20L + 0.39S + 0.24C + 48.7SS + 5ST + 2.6CP + 100O <= 8

4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP >= 4

9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP <= 200

0.6L + 0.6S - 0.4T - 0.4C - 0.4SS - 0.4ST -0.4CP - 0.40O >= 0

End

**Problem 4 Part B Output**

LP OPTIMUM FOUND AT STEP 3

OBJECTIVE FUNCTION VALUE

1) 1.554133

VARIABLE VALUE REDUCED COST

T 0.000000 1.002081

L 0.000000 0.402912

S 0.832298 0.000000

C 0.000000 0.486914

SS 0.096083 0.000000

ST 0.000000 0.405609

CP 1.152364 0.000000

O 0.000000 7.281258

ROW SLACK OR SURPLUS DUAL PRICES

2) 0.000000 -0.131261

3) 6.000000 0.000000

4) 0.000000 0.051847

5) 31.576324 0.000000

6) 55.651089 0.000000

7) 0.000000 -0.241358

NO. ITERATIONS= 3