Group 11

Charles Bennett

Conrad Lewin

Vlad Predovic

# CS 325 Project 3: Linear Programming

**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.*

The variables involved are defined as follows:

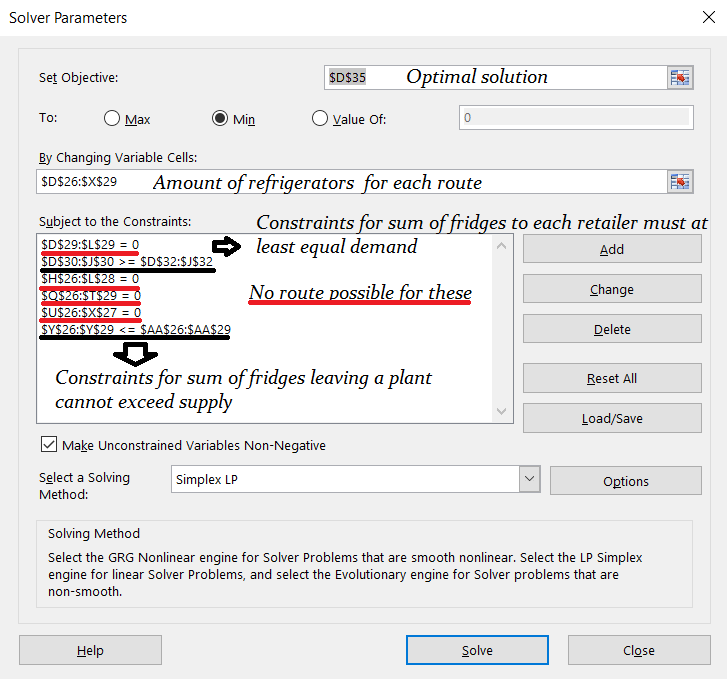
|  |  |
| --- | --- |
| Pi = Plant i where 0 < i 4.  Wj = Warehouse j where 0 < j 3.  Rk = Retailer k where 0 < k 7.  n = number of plants.  q = number of warehouses.  m = number of retailers. | cp(i,j) = Cost of shipping from Pi to Wj.  cw(j,k) = Cost of shipping from Wj to Rk.  si = max supply of plant i.  dk= demand of retailer k.  = fridges sent from Pi to Wj  = fridges sent from Wj to Rk |

The **objective function** is defined as follows:

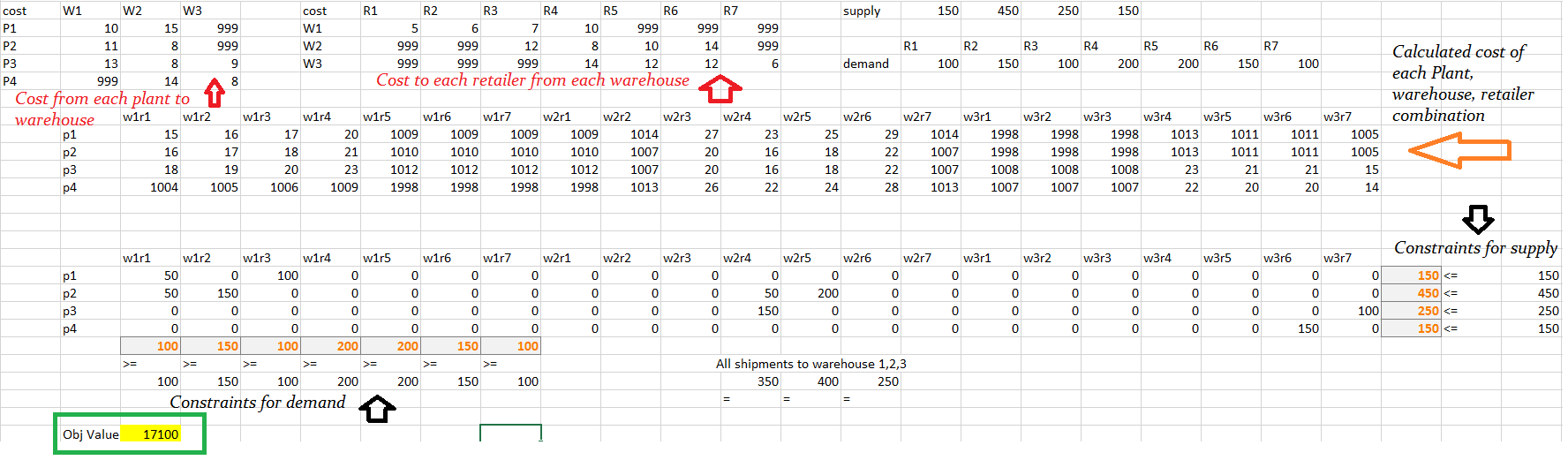
, where *Z* is sum of travel costs incurred by all the refrigerators.

**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.*



The excel solver was used for this problem. The optimal solution for minimizing the cost was found to be 17100 dollhairs. Infeasible routes were given a cost of 999 per unit to differentiate them for personal use. However these did not affect the overall solution due to the constraints forcing those routes to 0.   
Below can be found the spreadsheet used to create the values and constraints set in the excel solver window.



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

The minimum cost is $17100. The optimum routes are arranged by plant and then warehouse:

* Plant 1 ships 150 to warehouse 1, 50 goes to retailer 1 and 100 to 3.
* Plant 2 ships 200 to warehouse 1, 50 goes to retailer 1 and 150 to retailer 2
* Plant 2 ships 250 to warehouse 2, 50 goes to retailer 4 and 200 to retailer 5
* Plant 3 ships 150 to warehouse 2 and it all goes to retailer 4.
* Plant 3 ships 100 to warehouse 3 and it all goes to retailer 7.
* Plant 4 ships 150 to warehouse 3 and it all goes to retailer 6.

In other words the optimal routes are:

**From plant i to warehouse j:**

v(1,1), v(2,1), v(2,2), v(3,2), v(3,3), v(4,3)

**From warehouse j to retailer k:**

b(1,1), b(1,3), b(1,2), b(2,4), b(2,5), b(3,7), b(3,6).

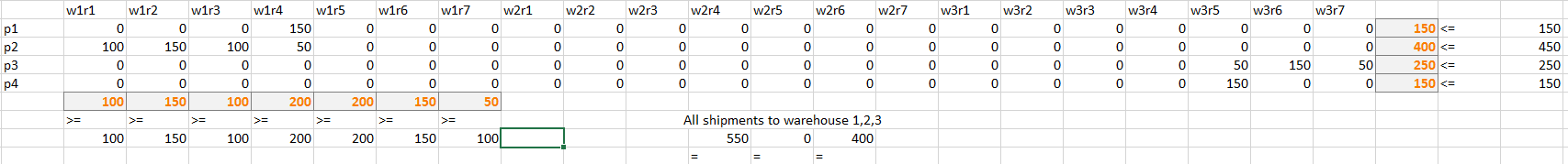
***Part B:*** *Due to old infrastructure Warehouse 2 is going to close eliminating all of the associated routes.What is the optimal solution for this modified model? Is it feasible to ship all the refrigerators to either warehouse 1 or 3 and then to the retailers without using warehouse 2? Why or why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

Running it in the excel solver resulted, the program was able to satisfy the request of every retailer until the seventh one as shown below (green box). From this detail the following was extracted:

By removing warehouse 2, only plants 3 and 4 are capable of reaching retailers 5, 6, and 7.

Plants 3 and 4 have a combined capacity of 400. However, the combined demand of the retailers is 450.

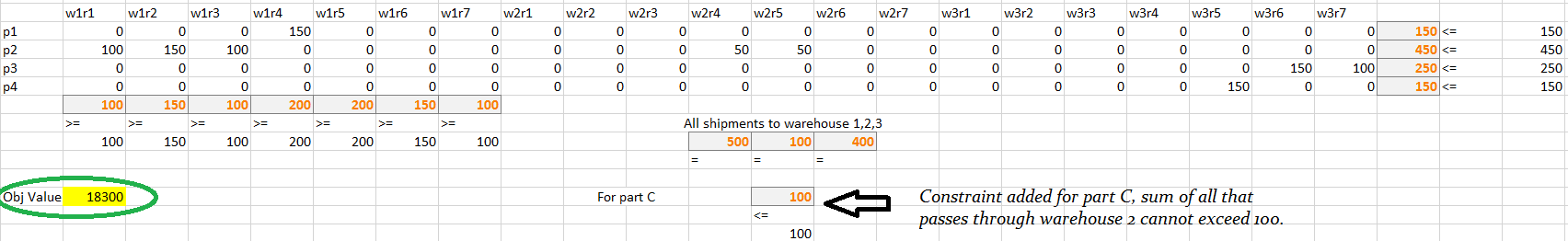
Therefore a solution is impossible.



***Part C:***  *Instead of closing Warehouse 2 management has decided 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?*

Yes there is a feasible solution with a cost of 18300. The routes are arranged by plant and then warehouse:

* Plant 1 ships 150 to warehouse 1 and it all goes to retailer 4.
* Plant 2 ships 350 to warehouse 1, 100 goes to retailer 1, 150 to retailer 2, 100 to retailer 3.
* Plant 2 ships 100 to warehouse 2, 50 goes to retailer 4, 50 to retailer 5.
* Plant 3 ships 250 to warehouse 3, 150 goes to retailer 6, 100 to retailer 7.
* Plant 4 ships 150 to warehouse 3 and it all goes to retailer 5



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

**Objective Function:**

, where *Z* is sum of travel costs incurred by all the refrigerators.

**Constraints:**

**Problem 2**

***Part A****: Determine the combination of ingredients that minimizes calories but meets all nutritional requirements.*

Allow for the following variable definitions:

v1 = 100 grams of tomato

v2 = 100 grams of lettuce

v3 = 100 grams of spinach

v4 = 100 grams of carrot

v5 = 100 grams of sunflower seed

v6 = 100 grams of smoked tofu

v7 = 100 grams of chickpeas

v8 = 100 grams of oil

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

**Objective Function:**

, where *K* is the minimum number of calories in a salad that meets all the nutritional requirements.

**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.*

The following equations were input into LINDO:

MIN 21 v1 + 16 v2 + 40 v3 + 41 v4 + 585 v5 + 120 v6 + 164 v7 + 884 v8

ST

0.85 v1 + 1.62 v2 + 2.86 v3 + 0.93 v4 + 23.4 v5 + 16.0 v6 + 9.00 v7 > 15

0.33 v1 + 0.20 v2 + 0.39 v3 + 0.24 v4 + 48.7 v5 + 5.00 v6 + 2.60 v7 + 100 v8 > 2

0.33 v1 + 0.20 v2 + 0.39 v3 + 0.24 v4 + 48.7 v5 + 5.00 v6 + 2.60 v7 + 100 v8 < 8

4.64 v1 + 2.37 v2 + 3.63 v3 + 9.58 v4 + 15.0 v5 + 3.00 v6 + 27.0 v7 > 4

9.00 v1 + 28.0 v2 + 65.0 v3 + 69.0 v4 + 3.80 v5 + 120 v6 + 78.0 v7 < 200

0.4 v1 - 0.4 v2 - 0.4 v3 - 0.4 v4 - 0.4 v5 - 0.4 v6 - 0.4 v7 - 0.4 v8 + v2 + v3 > 0

v1 > 0

v2 > 0

v3 > 0

v4 > 0

v5 > 0

v6 > 0

v7 > 0

v8 > 0

END

The optimal solution generated by LINDO:

|  |  |
| --- | --- |
| **Variable (ingredient)** | **Value (grams)** |
| v1 (tomato) |  |
| v2 (lettuce) |  |
| v3 (spinach) |  |
| v4 (carrot) |  |
| v5 (sunflower seeds) |  |
| v6 (smoked tofu) |  |
| v7 (chickpeas) |  |
| v8 (oil) |  |

The objective function returns a value of 114.7541. This represents the minimum number of calories needed to create a salad, a concoction containing 57 grams of lettuce and 87 grams of smoked tofu, that meets the nutritional requirements.

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

Since there are grams of lettuce and grams of smoked tofu in the salad, and both numbers represent a equivalent percentage of 100, we can sum the product of each weight with its price and derive the cost of the salad.

***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.*

Allow for the following variable definitions:

v1 = 100 grams of tomato

v2 = 100 grams of lettuce

v3 = 100 grams of spinach

v4 = 100 grams of carrot

v5 = 100 grams of sunflower seed

v6 = 100 grams of smoked tofu

v7 = 100 grams of chickpeas

v8 = 100 grams of oil

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

**Objective Function:**

, where *C* is the minimum cost of a salad that meets all the nutritional requirements.

**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.*

The following equations were input into LINDO:

MIN v1 + 0.75 v2 + 0.5 v3 + 0.5 v4 + 0.45 v5 + 2.15 v6 + 0.95 v7 + 2 v8

ST

0.85 v1 + 1.62 v2 + 2.86 v3 + 0.93 v4 + 23.4 v5 + 16.0 v6 + 9.00 v7 > 15

0.33 v1 + 0.20 v2 + 0.39 v3 + 0.24 v4 + 48.7 v5 + 5.00 v6 + 2.60 v7 + 100 v8 > 2

0.33 v1 + 0.20 v2 + 0.39 v3 + 0.24 v4 + 48.7 v5 + 5.00 v6 + 2.60 v7 + 100 v8 < 8

4.64 v1 + 2.37 v2 + 3.63 v3 + 9.58 v4 + 15.0 v5 + 3.00 v6 + 27.0 v7 > 4

9.00 v1 + 28.0 v2 + 65.0 v3 + 69.0 v4 + 3.80 v5 + 120 v6 + 78.0 v7 < 200

0.4 v1 - 0.4 v2 - 0.4 v3 - 0.4 v4 - 0.4 v5 - 0.4 v6 - 0.4 v7 - 0.4 v8 + v2 + v3 > 0

v1 > 0

v2 > 0

v3 > 0

v4 > 0

v5 > 0

v6 > 0

v7 > 0

v8 > 0

END

The optimal solution generated by LINDO:

|  |  |
| --- | --- |
| **Variable (ingredient)** | **Value (grams)** |
| v1 (tomato) |  |
| v2 (lettuce) |  |
| v3 (spinach) |  |
| v4 (carrot) |  |
| v5 (sunflower seeds) |  |
| v6 (smoked tofu) |  |
| v7 (chickpeas) |  |
| v8 (oil) |  |

The objective function returns a value of 1.554133 which is . This represents the minimum cost of a salad that meets the nutritional requirements. It contains 83 grams of lettuce, 9.6 grams of sunflower seeds and 115 grams of chickpeas.

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

The number of calories can be calculated by get the summing the products of each weight and its associated caloric value.

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 can select a combination of ingredients that is “near optimal” for both objectives. This is a type of multi-objective optimization.*

There seem to be three approaches Veronica can take to optimize her salad recipe:

1. She could derive the minimum number of ingredients by gram that her salad can contain while meeting all the nutritional requirements, with a cost no greater than $2.00 and a calorie count no more than 250.
2. She could deduce the minimum cost of a salad that meets all the nutritional requirements and is no more than 250 calories.
3. She could determine the minimum number of calories in a salad that meets all the nutritional requirements and costs no more than $2.00 to produce.

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

If she wants to maximize her profits, then she needs to find a salad recipe that is under 250 calories and is cheap to produce. In this case, her best bet is to take the second approach. Using LINDO to derive the optimal recipe we get:

|  |  |
| --- | --- |
| **Variable (ingredient)** | **Value (grams)** |
| v1 (tomato) |  |
| v2 (lettuce) |  |
| v3 (spinach) |  |
| v4 (carrot) |  |
| v5 (sunflower seeds) |  |
| v6 (smoked tofu) |  |
| v7 (chickpeas) |  |
| v8 (oil) |  |

If we round the previous values up, Veronica’s optimal salad recipe will require 76 grams of spinach, 9 grams of sunflower seeds, 17 grams of smoked tofu and 88 grams of chickpeas. Such a salad will contain

and the production cost will be

If she chooses to sell this salad at a price of $5.00, then she will make a profit of $3.38 for ever salad sold. Furthermore, she will make more sales since the salad contains less than 250 calories.

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

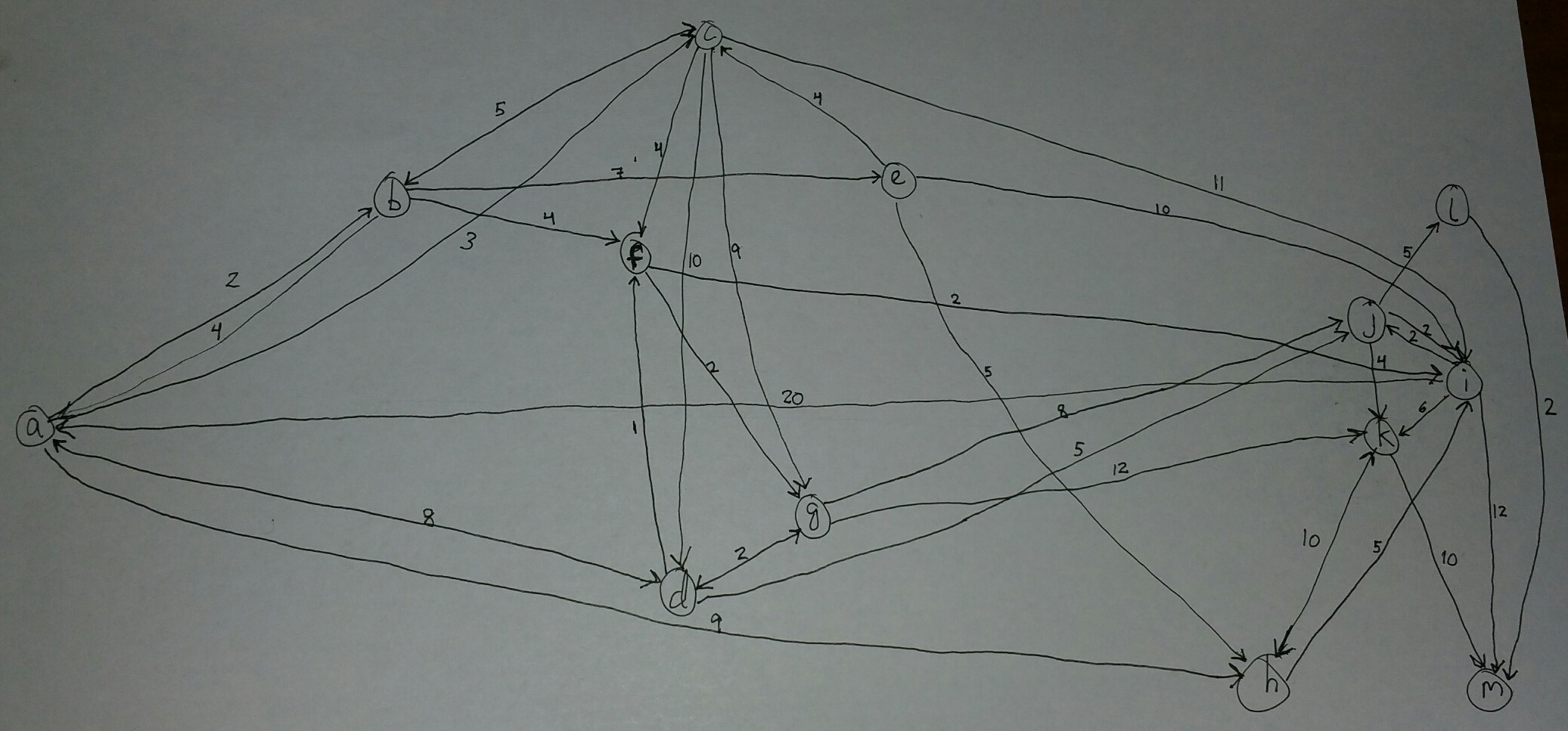
The previous solution was derived by applying each optimization approach and comparing the results. Since each approach produces a salad that meets all the necessary constraints, the “best” salad will be the one that creates the most profit. Below is a table of the salads created using each approach (LINDO):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach** | **Salad Description (grams rounded up)** | **Calories** | **Cost** | **Profit** |
| 1 | 53g spinach, 9g sunflower seeds, 72g smoked tofu | 160.52 kcal | $1.85 | $3.15 |
| 2 | 76g spinach, 9g sunflower seeds, 17g smoked tofu, 88g chickpeas | 247.77 kcal | $1.62 | $3.38 |
| 3 | 55g spinach, 3g sunflower seeds, 80g smoked tofu | 135.55 kcal | $2.00 | $3.00 |

Clearly, the recipe derived from approach 2 will net Veronica the largest profits.

**Problem 3**

Rough sketch of the directed graph:



***Part A:*** *What are the lengths of the shortest paths from vertex a to all other vertices?*

**The problem:**

Maximize d(x)  
Subject to d(a) = 0

d(v)-d(u) <= l(u→v) for every edge u→v

**Code** (replace “max db” with “max dc”, “max dd”, etc to find the distance from a to each vertex):

max db

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

Shortest lengths from vertex a to each other vertex:

|  |  |  |
| --- | --- | --- |
| **Start Vertex** | **End Vertex** | **Distance** |
| a | a | 0 |
| a | b | 2 |
| a | c | 3 |
| a | d | 8 |
| a | e | 9 |
| a | f | 6 |
| a | g | 8 |
| a | h | 9 |
| a | i | 8 |
| a | j | 10 |
| a | k | 14 |
| a | l | 15 |
| a | m | 17 |

***Part 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)?*

An “UNBOUNDED” error will occur, and the distance will approach infinity.

***Part 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?*

To find a path from all vertices to m using one linear program, we can reverse the directions of the edges that represent the distances between vertices. Swap v and u for every statement after ST (such that). This reverses the direction the path travels, going from m to each vertex, but this method still gives the correct distances because although the directions are reversed, the order of visited vertices and their respective distances stay the same.

**Example:**

Maximize d(x)  
Subject to d(m) = 0

d(u)-d(v) <= l(v→u) for every edge v→u

**Code** (replace “max da” with “max db”, “max dc”, etc to find the distance from each vertex to m):

max da

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

END

Shortest lengths from each vertex to vertex m:

|  |  |  |
| --- | --- | --- |
| **Start Vertex** | **End Vertex** | **Distance** |
| a | m | 17 |
| b | m | 15 |
| c | m | 15 |
| d | m | 12 |
| e | m | 19 |
| f | m | 11 |
| g | m | 14 |
| h | m | 14 |
| i | m | 9 |
| j | m | 7 |
| k | m | 10 |
| l | m | 2 |
| m | m | 0 |

***Part 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).*

To find the length of the shortest path from any vertex to vertex y that passes through vertex i, you can use two linear programs. One linear program finds the distance from x to i, and the second linear program finds the distance from i to y. The results of each linear program can be combined to give us the total distance.

**Code** (replace da=0 with db=0, dc=0, etc to find the shortest length from each vertex to i):

From a to i:

max di

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

**Code** (replace max da with max db, max dc, etc to find the shortest length from i to each vertex):

**From i to a:**

max da

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

We combine the distance from the starting vertex to the middle vertex i with the distance from vertex i to the end vertex, giving us the total distance as shown in the tables below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| a | i | 8 | a | 20 | 28 |
| a | i | 8 | b | 22 | 30 |
| a | i | 8 | c | 23 | 31 |
| a | i | 8 | d | 28 | 36 |
| a | i | 8 | e | 29 | 37 |
| a | i | 8 | f | 26 | 34 |
| a | i | 8 | g | 28 | 36 |
| a | i | 8 | h | 16 | 24 |
| a | i | 8 | i | 0 | 8 |
| a | i | 8 | j | 2 | 10 |
| a | i | 8 | k | 6 | 14 |
| a | i | 8 | l | 7 | 15 |
| a | i | 8 | m | 9 | 17 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| b | i | 6 | a | 20 | 26 |
| b | i | 6 | b | 22 | 28 |
| b | i | 6 | c | 23 | 29 |
| b | i | 6 | d | 28 | 34 |
| b | i | 6 | e | 29 | 35 |
| b | i | 6 | f | 26 | 32 |
| b | i | 6 | g | 28 | 34 |
| b | i | 6 | h | 16 | 22 |
| b | i | 6 | i | 0 | 6 |
| b | i | 6 | j | 2 | 8 |
| b | i | 6 | k | 6 | 12 |
| b | i | 6 | l | 7 | 13 |
| b | i | 6 | m | 9 | 15 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| c | i | 6 | a | 20 | 26 |
| c | i | 6 | b | 22 | 28 |
| c | i | 6 | c | 23 | 29 |
| c | i | 6 | d | 28 | 34 |
| c | i | 6 | e | 29 | 35 |
| c | i | 6 | f | 26 | 32 |
| c | i | 6 | g | 28 | 34 |
| c | i | 6 | h | 16 | 22 |
| c | i | 6 | i | 0 | 6 |
| c | i | 6 | j | 2 | 8 |
| c | i | 6 | k | 6 | 12 |
| c | i | 6 | l | 7 | 13 |
| c | i | 6 | m | 9 | 15 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| d | i | 3 | a | 20 | 23 |
| d | i | 3 | b | 22 | 25 |
| d | i | 3 | c | 23 | 26 |
| d | i | 3 | d | 28 | 31 |
| d | i | 3 | e | 29 | 32 |
| d | i | 3 | f | 26 | 29 |
| d | i | 3 | g | 28 | 31 |
| d | i | 3 | h | 16 | 19 |
| d | i | 3 | i | 0 | 3 |
| d | i | 3 | j | 2 | 5 |
| d | i | 3 | k | 6 | 9 |
| d | i | 3 | l | 7 | 10 |
| d | i | 3 | m | 9 | 12 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| e | i | 10 | a | 20 | 30 |
| e | i | 10 | b | 22 | 32 |
| e | i | 10 | c | 23 | 33 |
| e | i | 10 | d | 28 | 38 |
| e | i | 10 | e | 29 | 39 |
| e | i | 10 | f | 26 | 36 |
| e | i | 10 | g | 28 | 38 |
| e | i | 10 | h | 16 | 26 |
| e | i | 10 | i | 0 | 10 |
| e | i | 10 | j | 2 | 12 |
| e | i | 10 | k | 6 | 16 |
| e | i | 10 | l | 7 | 17 |
| e | i | 10 | m | 9 | 19 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| f | i | 2 | a | 20 | 22 |
| f | i | 2 | b | 22 | 24 |
| f | i | 2 | c | 23 | 25 |
| f | i | 2 | d | 28 | 30 |
| f | i | 2 | e | 29 | 31 |
| f | i | 2 | f | 26 | 28 |
| f | i | 2 | g | 28 | 30 |
| f | i | 2 | h | 16 | 18 |
| f | i | 2 | i | 0 | 2 |
| f | i | 2 | j | 2 | 4 |
| f | i | 2 | k | 6 | 8 |
| f | i | 2 | l | 7 | 9 |
| f | i | 2 | m | 9 | 11 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| g | i | 5 | a | 20 | 25 |
| g | i | 5 | b | 22 | 27 |
| g | i | 5 | c | 23 | 28 |
| g | i | 5 | d | 28 | 33 |
| g | i | 5 | e | 29 | 34 |
| g | i | 5 | f | 26 | 31 |
| g | i | 5 | g | 28 | 33 |
| g | i | 5 | h | 16 | 21 |
| g | i | 5 | i | 0 | 5 |
| g | i | 5 | j | 2 | 7 |
| g | i | 5 | k | 6 | 11 |
| g | i | 5 | l | 7 | 12 |
| g | i | 5 | m | 9 | 14 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| h | i | 5 | a | 20 | 25 |
| h | i | 5 | b | 22 | 27 |
| h | i | 5 | c | 23 | 28 |
| h | i | 5 | d | 28 | 33 |
| h | i | 5 | e | 29 | 34 |
| h | i | 5 | f | 26 | 31 |
| h | i | 5 | g | 28 | 33 |
| h | i | 5 | h | 16 | 21 |
| h | i | 5 | i | 0 | 5 |
| h | i | 5 | j | 2 | 7 |
| h | i | 5 | k | 6 | 11 |
| h | i | 5 | l | 7 | 12 |
| h | i | 5 | m | 9 | 14 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| i | i | 0 | a | 20 | 20 |
| i | i | 0 | b | 22 | 22 |
| i | i | 0 | c | 23 | 23 |
| i | i | 0 | d | 28 | 28 |
| i | i | 0 | e | 29 | 29 |
| i | i | 0 | f | 26 | 26 |
| i | i | 0 | g | 28 | 28 |
| i | i | 0 | h | 16 | 16 |
| i | i | 0 | i | 0 | 0 |
| i | i | 0 | j | 2 | 2 |
| i | i | 0 | k | 6 | 6 |
| i | i | 0 | l | 7 | 7 |
| i | i | 0 | m | 9 | 9 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| j | i | 2 | a | 20 | 22 |
| j | i | 2 | b | 22 | 24 |
| j | i | 2 | c | 23 | 25 |
| j | i | 2 | d | 28 | 30 |
| j | i | 2 | e | 29 | 31 |
| j | i | 2 | f | 26 | 28 |
| j | i | 2 | g | 28 | 30 |
| j | i | 2 | h | 16 | 18 |
| j | i | 2 | i | 0 | 2 |
| j | i | 2 | j | 2 | 4 |
| j | i | 2 | k | 6 | 8 |
| j | i | 2 | l | 7 | 9 |
| j | i | 2 | m | 9 | 11 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| k | i | 15 | a | 20 | 35 |
| k | i | 15 | b | 22 | 37 |
| k | i | 15 | c | 23 | 38 |
| k | i | 15 | d | 28 | 43 |
| k | i | 15 | e | 29 | 44 |
| k | i | 15 | f | 26 | 41 |
| k | i | 15 | g | 28 | 43 |
| k | i | 15 | h | 16 | 31 |
| k | i | 15 | i | 0 | 15 |
| k | i | 15 | j | 2 | 17 |
| k | i | 15 | k | 6 | 21 |
| k | i | 15 | l | 7 | 22 |
| k | i | 15 | m | 9 | 24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| l | i | Unbounded | a | 20 | Unbounded |
| l | i | Unbounded | b | 22 | Unbounded |
| l | i | Unbounded | c | 23 | Unbounded |
| l | i | Unbounded | d | 28 | Unbounded |
| l | i | Unbounded | e | 29 | Unbounded |
| l | i | Unbounded | f | 26 | Unbounded |
| l | i | Unbounded | g | 28 | Unbounded |
| l | i | Unbounded | h | 16 | Unbounded |
| l | i | Unbounded | i | 0 | Unbounded |
| l | i | Unbounded | j | 2 | Unbounded |
| l | i | Unbounded | k | 6 | Unbounded |
| l | i | Unbounded | l | 7 | Unbounded |
| l | i | Unbounded | m | 9 | Unbounded |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Start Vertex** | **Mid Vertex** | **Distance Start to Mid** | **End Vertex** | **Distance Mid to End** | **Total Distance** |
| m | i | Unbounded | a | 20 | Unbounded |
| m | i | Unbounded | b | 22 | Unbounded |
| m | i | Unbounded | c | 23 | Unbounded |
| m | i | Unbounded | d | 28 | Unbounded |
| m | i | Unbounded | e | 29 | Unbounded |
| m | i | Unbounded | f | 26 | Unbounded |
| m | i | Unbounded | g | 28 | Unbounded |
| m | i | Unbounded | h | 16 | Unbounded |
| m | i | Unbounded | i | 0 | Unbounded |
| m | i | Unbounded | j | 2 | Unbounded |
| m | i | Unbounded | k | 6 | Unbounded |
| m | i | Unbounded | l | 7 | Unbounded |
| m | i | Unbounded | m | 9 | Unbounded |