Math381 Assignment 1

Yuchen AN

June 28, 2023

1 Introduction

The knapsack problem is a well-known combination optimization problem. Typically, in the knapsack problem, we need to pack a set of items, with given values and sizes (such as weights or volumes), into a container with a maximum capacity.
In this assignment, we try to solve a knapsack problem. Suppose we have a container which has weight capacity 100kg and volume capacity 100 liters. From 100 unique object with various volumes V, weights W, and volumes P, we try to figure out what objects should be put into this container to maximize the total value.

For object i($1 \leq i \leq 100$)
value = v_i = floor(3+27cos^2(13i))dollars
weight = w_i = floor(4+38cos^2(12i+5))kg
volume= p_i = floor(4+28cos^2(7i+8))liters

2 Part a

In this part, I will find out what objects should be put into the container to get the maximum value. The weight capacity is 100kg. The volume capacity is 100 liters. x_i is a binary variable which indicates weather or not the object i is put in to the container.

$x_i = 0$ if object i is not in the container.
$x_i = 1$ if object i is in the container.

Here is the mathematical formulation of LP I am going to solve:

Maximize $\sum_{n=1}^{100} v_i\, x_i$(The objective function); subject to
$\sum_{n=1}^{100} w_i\, x_i \leq 100$ (The sum of weights of all objects cannot greater than 100 kg);
$\sum_{n=1}^{100} v_i x_i \leq 100$ (The sum of volumes of all objects cannot greater tha 100 liters);
$x_i \in \{0, 1\}$, $1 \leq i \leq 100$

Firstly, I use Python to create the lpsolve input file. The following code helps to form this LP problem.

```
# The following code generates a lpsolve input file.
# This file determines what objects should be put into the container to maximize the total value.

import math

#  To get the objective Function
fullString = " "
for i in range(1, 101):
    value = math.floor(3 + 27 * math.cos(13*i)**2)
    fullString = fullString + "+" + str(value) + "*x_" + str(i)
print("max: "+fullString+";")

# The weight capacity is 100
weight_constraint = " "
```

```
for i in range(1,101):
    weight = math.floor( 4 + 38*math.cos(12*i + 5)**2)
    weight_constraint = weight_constraint + "+" + str(weight) + "*x_" + str(i)
print(weight_constraint + "<= 100;")

# The volume capacity is 100
volume_constraint = " "
for i in range(1,101):
    volume = math.floor( 4 + 28*math.cos(7*i - 8)**2)
    volume_constraint = volume_constraint + "+" + str(volume) + "*x_" + str(i)
print(volume_constraint + "<= 100;")

# declare all variables as binary
binString = "bin x_1"
for i in range(2,101):
    binString = binString + ",x_" + str(i)
print(binString + ";")
```

By writing related codes in Python, I get the lpsolve input file looks like this:

```
max: +25x_1+14x_2+4x_3+......+3x_98+9x_99+20x_100;
(The following line ensure the total weight of all objects less than the 100kg)
weight=+6x_1+25x_2+41x_3+......+39xx_98+22x_99+5x_100;
weight<=100;
(The following line ensure the total volume of all objects less than the 100 liters)
volume=+12x_1+29x_2+27x_3+......+31x_99+16x_100;
volume<=100;
(The following line ensure all variables are binary)
bin x_1,x_2,.......,x_99,x_100
```

Here is the output generated by the lpsolver. For the purpose of readability, we only include the objects that are put into the container:

```
Value of objective function: 245.00000000

Actual values of the variables:
x_1                             1
x_22                            1
x_35                            1
x_44                            1
x_45                            1
x_57                            1
x_66                            1
x_78                            1
x_79                            1
x_95                            1
weight                        100
volume                         96

All other variables are zero.
```

We can see that object 1, 22, 35, 44, 45, 57, 66, 78, 79, 95 can be put into the container to maximize the total value. The maximum possible total value is 245. From the calculation, we can get the sum of weights of that ten objects is 100. The sum of volumes of that ten objects is 96. We know that the weight capacity is 100kg, and the volume capacity is 100 liters. The weight constraint is binding since if we replace the 100 in the constraint with any smaller number, the solution would change. The volume constraint is not binding since 96 is less than 100.

3 Part b

For part b, we want to explore whether the optimal solutions will change if we add lower bound on the number of objects that we put into the container. Since we have already got the optimal solutions for the original LP, which are object 1, 22, 35, 44, 45, 57, 66, 78, 79, 95. There are total 10 object should be put into the container. Thus, when we set the lower bound on the number of objects put into the container, we do not consider the bound less than 10. Because the optimal solutions will not change if the the bound is less than 10. We add an additional constraint for the original LP: Let b represents the lower bound on the number of objects that we put into the container, b>10

$$\sum_{n=1}^{100} x_i \geq \text{b}$$

We start with b=11. And then we try b=12, b=13, and so on. However, when we go up to b=14, we get the output: This problem is infeasible. we know that a linear program is infeasible if there exists no solution that satisfies all of the constraints. Therefore, 13 is the largest lower bound we put on the the number of objects that we put into the container. We conclude that there is a solution for the LP if $0 \leq b \leq 13$.
After adding the constraint into lpsolve and changing the lower bound in each case, I make a table to compare the optimal solutions from different lower bounds. The leftmost column represents the lower bound. The second column is the total value of objects put in the container. Other columns represent objects that we put into the container in each case.

```
10 245  1       22     35       44 45 57    66    78 79       95
11 243  1       22 23           44 45 57    66    78 79    84    100
12 237  1 17    22 23           44 45 57    62 66    79    84    100
13 177  1    18 22 23     39 40 44 45    61 62       79 83 84
```

From the table, we find that the number of objects becomes more as the lower bound increases. Object 1, 22, 44, 45, 79 are in more than one solution. I made a table to compare their weights, volumes, values.

object	value	weight	volume
1	25	6	12
22	29	9	4
44	28	12	4
45	19	4	15
79	27	11	4

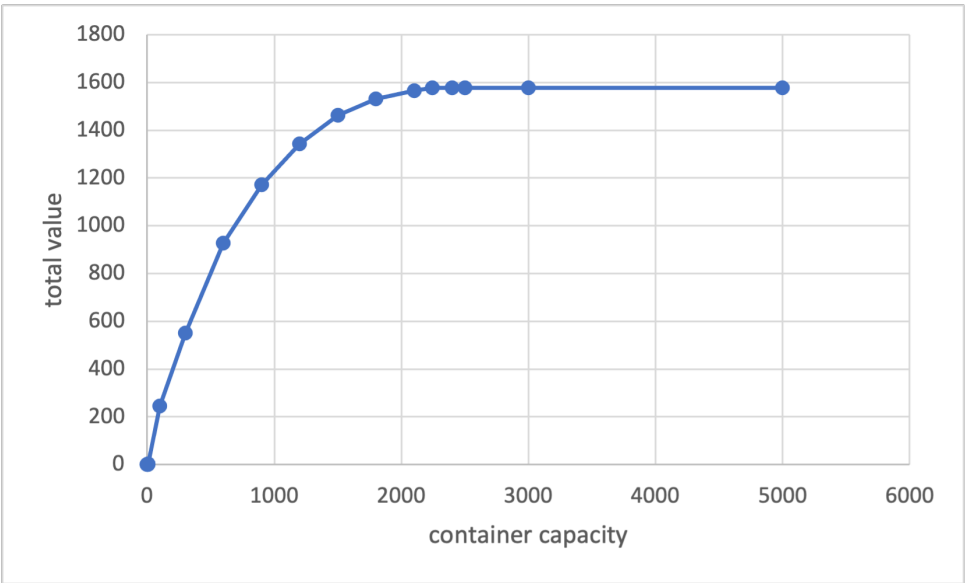
From the table, we can see the volume of object 22, 44, 79 is 4 liters, which is very small. Object 45 only weight 4kg, which is light. Those 5 objects all have high values, so they are included in all solutions.

4 Part c

For part c, we want to find out how the value change if we change the weight and volume bounds and make weight bound and volume bound numerically equal. Let x be the bounds of weight and volume. Total value means the sum of value of all objects in the container. The following table shows the bounds and corresponding values.

x	Total Value
0	0
3	0
5	0
6	3
100	245
300	551
600	927
900	1172
1200	1342
1500	1462
1800	1532
2100	1566
2244	1578
2400	1578
2500	1578
3000	1578
5000	1578

Then, we make a graph according to the table by using the excel, which better clarify trends.



From the graph, we can see that the total value increases as the container capacity increases. When the container capacity is 6, the total value becomes 6 instead of 0. This means that we can put some object into the container. As the container capacity increases, the graph becomes increasingly flatter, which means the increase rate is getting slower. After container capacity reaches to 2244, the total value stops increasing. As the container capacity change from 2244 to 5000, the total value remains constant as 1578. Therefore, the total value will not change if we have a container with wight capacity and volume capacity being 2244 or larger. One reason may be that when the container capacity is 2244, it can hold all objects.