Assignment 1

Christian Diangco

MATH 381 A

1. Suppose we are a band of pirates that have stumbled upon a treasure room.

In this treasure room there are 50 unique items.

Item , where , has a value, weight, and volume given by:

= thousands of dollars

= kg

= liters

In order to decide which items we will take, we will use the variable which can either be equal to 1 or 0.

1 means we take the item, 0 means we do not.

Our treasure chest has a max weight capacity of 200 kg and volume capacity of 100 liters.

We are trying to select the items that maximize the total value while obeying the weight and volume limits of our treasure chest.

We will use an LP to solve this problem.

To maximize the total value of the items we take, our objective function will be the summation of all 50 item values multiplied by .

We must obey the weight and volume limits of our treasure chest so the total weight and volume of the items we take must be less than or equal to 200 kg and 100 liters respectively.

These constraints determine whether will be 1 or 0 for each value of *i*, telling us which items we will be taking.

To create the input file for LPSolve we will be using this Python code:

from math import \*

# This is the input file for the LP solver that we will be writing to.

f = open("Assignment1.txt", "w")

# This line contains the function we are trying to maximize. In this case, it is for the values of the objects.

obj\_function = "max: "

# This line contains the weight constraint. The container has a max weight of 200 kg.

weight\_constraint = ""

# This line contains the volume constraint. The container has a max volume of 100 liters.

volume\_constraint = ""

# This line contains each of the binary variables.

variables = "bin "

# This loop fills in each of the lines above to then be written into the text file.

for i in range(1, 51):

obj\_function += "+" + str(floor(50 + 30\*cos(i))) + "x" + str(i)

weight\_constraint += "+" + str(floor(14 + 9\*cos(11\*i + 2))) + "x" + str(i)

volume\_constraint += "+" + str(floor(10 + 2\*cos(4\*i - 1))) + "x" + str(i)

variables += "x" + str(i) + ","

obj\_function += ";\n"

weight\_constraint += " <= 200;\n"

volume\_constraint += " <= 100;\n"

variables = variables[:-1]

variables += ";\n"

f.writelines([obj\_function, weight\_constraint, volume\_constraint, variables])

f.close()

Using this code, the generated input file looks like this:

max: +66x1+37x2+20x3+30x4+58x5+78x6+72x7+45x8+22x9+24x10+50x11+75x12+77x13+54x14+27x15+21x16+41x17+69x18+79x19+62x20+33x21+20x22+34x23+62x24+79x25+69x26+41x27+21x28+27x29+54x30+77x31+75x32+49x33+24x34+22x35+46x36+72x37+78x38+57x39+29x40+20x41+38x42+66x43+79x44+65x45+37x46+20x47+30x48+59x49+78x50;

+22x1+17x2+5x3+10x4+22x5+17x6+5x7+9x8+22x9+18x10+6x11+9x12+21x13+18x14+6x15+9x16+21x17+18x18+6x19+9x20+21x21+18x22+6x23+9x24+21x25+18x26+6x27+9x28+21x29+18x30+6x31+9x32+21x33+18x34+6x35+9x36+21x37+19x38+6x39+8x40+21x41+19x42+6x43+8x44+21x45+19x46+6x47+8x48+21x49+19x50 <= 200;

+8x1+11x2+10x3+8x4+11x5+8x6+9x7+11x8+8x9+10x10+11x11+8x12+11x13+10x14+8x15+11x16+8x17+9x18+11x19+8x20+10x21+11x22+8x23+11x24+10x25+8x26+11x27+8x28+9x29+11x30+8x31+10x32+11x33+8x34+11x35+10x36+8x37+11x38+9x39+9x40+11x41+8x42+10x43+11x44+8x45+11x46+10x47+8x48+11x49+9x50 <= 100;

bin x1,x2,... ,x50;

After LPSolve with this input file, this is the result:

Value of the objective function: 840.00000000

Actual values of the variables:

x1 1

x6 1

x7 1

x12 1

x18 1

x20 1

x26 1

x31 1

x37 1

x39 1

x45 1

x50 1

All other variables are zero.

The items taken to maximize the total value are:

1, 6, 7, 12, 18, 20, 26, 31, 37, 39, 45, 50

The total value of the items we’ve taken is $840,000.

1. Now let’s suppose that the pirate captain wants the value of prime-indexed items to make up at least 50% of the total value of the items we take.

We will again use an LP to solve this problem.

The LP will be exactly the same as in part a but with this additional constraint:

We can make some additions to our code in order to add this constraint to the LPSolve input file.

from math import \*

# Returns whether a number is prime or not.

def is\_prime(n):

if n == 1:

return False

for i in range(2, int(n/2) + 1):

if n % i == 0:

return False

return True

# This is the input file for the LP solver that we will be writing to.

f = open("Assignment1.txt", "w")

# This line contains the function we are trying to maximize. In this case, it is for the values of the objects.

obj\_function = "max: "

# This line contains the weight constraint. The container has a max weight of 200 kg.

weight\_constraint = ""

# This line contains the volume constraint. The container has a max volume of 100 liters.

volume\_constraint = ""

# This line contains each of the binary variables.

variables = "bin "

# This line contains the constraint where the value of prime-indexed items must make up at least 50% of the total value.

prime\_constraint = ""

# This contains the value for each variable divided by 2, which will go on the right side of the prime index constraint.

prime\_constraintRHS = ""

# This loop fills in each of the lines above to then be written into the text file.

for i in range(1, 51):

obj\_function += "+" + str(floor(50 + 30\*cos(i))) + "x" + str(i)

weight\_constraint += "+" + str(floor(14 + 9\*cos(11\*i + 2))) + "x" + str(i)

volume\_constraint += "+" + str(floor(10 + 2\*cos(4\*i - 1))) + "x" + str(i)

variables += "x" + str(i) + ","

prime\_constraintRHS += "+" + str(floor((50 + 30\*cos(i)))/2) + "x" + str(i)

if is\_prime(i):

prime\_constraint += "+" + str(floor(50 + 30\*cos(i))) + "x" + str(i)

obj\_function += ";\n"

weight\_constraint += " <= 200;\n"

volume\_constraint += " <= 100;\n"

variables = variables[:-1]

variables += ";\n"

prime\_constraint += " >= " + prime\_constraintRHS + ";\n"

f.writelines([obj\_function, weight\_constraint, volume\_constraint, prime\_constraint, variables])

f.close()

With this code, we add this line to the LPSolve input file:

+37x2+20x3+58x5+72x7+50x11+77x13+41x17+79x19+34x23+27x29+77x31+72x37+20x41+66x43+20x47 >= +33.0x1+18.5x2+10.0x3+15.0x4+29.0x5+39.0x6+36.0x7+22.5x8+11.0x9+12.0x10+25.0x11+37.5x12+38.5x13+27.0x14+13.5x15+10.5x16+20.5x17+34.5x18+39.5x19+31.0x20+16.5x21+10.0x22+17.0x23+31.0x24+39.5x25+34.5x26+20.5x27+10.5x28+13.5x29+27.0x30+38.5x31+37.5x32+24.5x33+12.0x34+11.0x35+23.0x36+36.0x37+39.0x38+28.5x39+14.5x40+10.0x41+19.0x42+33.0x43+39.5x44+32.5x45+18.5x46+10.0x47+15.0x48+29.5x49+39.0x50;

After using LPSolve with this input file, we get the following output:

Value of objective function: 822.00000000

Actual values of the variables:

x6 1

x7 1

x12 1

x13 1

x19 1

x25 1

x26 1

x31 1

x37 1

x43 1

x50 1

All other variables are zero.

The items taken to maximize the total value are:

6, 7, 12, 13, 19, 25, 26, 31, 37, 43, 50

The items we selected have a total value of $822,000.

Compared to our results from part a, our total value is lower and we selected only 11 items instead of 12.

This selection also contains more prime-indexed items.

We went from 3 prime-indexed items selected to 6.

Now let’s suppose that the pirate captain wants the value of prime-indexed items to make up at least 75% of the total value of the items we take rather than 50%.

With this, our prime-index constraint changes to

To change the constraint in our LP input file, we just need to change one line of our code:

prime\_constraintRHS += "+" + str(floor((50 + 30\*cos(i)))\*3/4) + "x" + str(i)

With this change, our prime-index constraint in our LPSolve input file now looks like this:

+37x2+20x3+58x5+72x7+50x11+77x13+41x17+79x19+34x23+27x29+77x31+72x37+20x41+66x43+20x47 >= +49.5x1+27.75x2+15.0x3+22.5x4+43.5x5+58.5x6+54.0x7+33.75x8+16.5x9+18.0x10+37.5x11+56.25x12+57.75x13+40.5x14+20.25x15+15.75x16+30.75x17+51.75x18+59.25x19+46.5x20+24.75x21+15.0x22+25.5x23+46.5x24+59.25x25+51.75x26+30.75x27+15.75x28+20.25x29+40.5x30+57.75x31+56.25x32+36.75x33+18.0x34+16.5x35+34.5x36+54.0x37+58.5x38+42.75x39+21.75x40+15.0x41+28.5x42+49.5x43+59.25x44+48.75x45+27.75x46+15.0x47+22.5x48+44.25x49+58.5x50;

Here are the results from using LPSolve with this input file after changing the constraint:

Value of objective function: 729.00000000

Actual values of the variables:

x5 1

x6 1

x7 1

x12 1

x13 1

x17 1

x19 1

x23 1

x31 1

x37 1

x43 1

All other variables are zero.

The items taken to maximize the total value are:

5, 6, 7, 12, 13, 17, 19, 23, 31, 37, 43

The total value of the items selected has further decreased from $822,000 to $729,000.

We have again selected 11 items but this time with even more prime-indexed items, going from 6 to 9.

1. Now let’s suppose that we brought a cloning machine to the treasure room and again want to maximize the total value of the items we take.

We can take as many of each item as we want while obeying the constraints of our treasure chest.

It must be noted that we are no longer considering the prime-index constraint from part b.

To accomplish this, we just need to get rid of the prime-index constraint from our LPSolve input file and change our variables from binary to integers like so:

int x1,x2,... ,x50;

After using LPSolve with this input file we receive the following output:

Value of objective function: 937.00000000

Actual values of the variables:

x6 10

x19 1

x50 1

To maximize the total value, the items taken were:

10 of item 6,

1 of item 19,

1 of item 50

The total value for the items selected is $937,000, the highest total value so far.

We also selected only three unique items.

Notably, item 6 was chosen 10 times.

This is because of its high value of $78,000 while having a weight of 17 kg and volume of 8 liters.

Items 19 and 50 also have high values, those being $79,000 and $78,000 respectively.

However, item 19 has more volume than item 6 and item 50 has both more weight and volume than item 6.

This is why only 1 of each of these items were taken.