Writing Assignment 2 - MATH 381

Brian Kang

January 18, 2019

Suppose you have the following unique objects, each with a weight, a volume and a value, as shown.

object	weight	volume	value
1	10	5	72
2	11	6	80
3	7	10	66
4	12	4	75
5	3	9	68
6	4	8	47
7	9	12	85
8	2	10	55
9	8	8	82

Which objects should you put in a knapsack with weight capacity 40 and volume capacity 45 such that the total value is maximized?

Define an IP for this problem. Describe it mathematically. Then solve it with lpsolve. Include all lpsolve input and output, and comment on the solution (i.e., don't just give the output of lpsolve). Are the constraints binding or not?

Then, solve the problem again, this time assuming that the list above gives a list of types of objects, and there is an unlimited number of each object type available. What should go in your knapsack now?

1.

The first part of the IP problem to solve here is the traditional knapsack problem. We aim to maximize the total value of the objects (items) we take, but constrained to the total weight and volume our knapsack can withstand. Note that there is only one of each item, so our decisions for each item is to either take it or not take it. This can be illustrated by defining a binary variable

$$x_i \in \{0, 1\}$$

for each object $i \in \{1, \ldots, 9\}$, where $x_i = 0$ if we do not put item i in the knapsack and $x_i = 1$ if we do. Our objective function is then,

$$\max \sum_{i=1}^{9} k_i x_i$$

where k_i corresponds to the value of the object i. Our two constraints are then,

$$\textstyle\sum_{i=1}^{9} w_i x_i \leq 40$$

$$\textstyle\sum_{i=1}^{9} v_i x_i \leq 45$$

where w_i is the weight of the object i and v_i is the volume of the object i. Total weight of the items we put must be less than or equal to 40 and volume must be less that or equal to 45. Solving this IP can be done through inputting the below code to LPSolve.

```
/* Objective function */
max: 72x_1+80x_2+66x_3+75x_4+68x_5+47x_6+85x_7+55x_8+82x_9;

/* Variable bounds */
c1: +10x_1+11x_2+7x_3+12x_4+3x_5+4x_6+9x_7+2x_8+8x_9 <= 40;
c2: +5x_1+6x_2+10x_3+4x_4+9x_5+8x_6+12x_7+10x_8+8x_9 <= 45;

/* Variables */
bin x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9;
```

From the input, our objective function is the maximum sum of the products of each object's values and the decision of we take it or not. And our constrains are the sum of the products of the each object that we take and their corresponding weight and volume. Each respective sums of products is limited to 40 and 45 units, as already stated. We can see that the x_i variables are declared binary as we desired to.

The output we get from the LPSolve is the following,

Variables	result
objective	407
x_1	0
x_2	1
x_3	0
x_4	1
x_5	1
x_6	1
x_7	0
x_8	1
x_9	1

The output tells us that the max total value achievable with the constraints satisfied is 407. The objects we put in the knapsack are then objects 2,4,5,6,8, and 9. (Note: from my Windows machine, a ThinkPad T470s, the entire process took 0.017 seconds.) We can verify the constraints by plugging the objects back into the constraint. For the first constraint, summing the weights of the objects we take,

$$11 + 12 + 3 + 4 + 2 + 8 = 40 \leq 40$$

is indeed true, and for the second constraint, likewise,

$$6 + 4 + 9 + 8 + 10 + 8 = 45 \leq 45$$

is also true. Since the left and right side of both constraints are equal, we can conclude that both constraints are binding.

2.

The second part of the problem to solve is a variation to the tradition knapsack problem we just solved. This time, we assume that the objects we are given are actually the types of objects, i.e., we are given the information for object type 1, type 2, and so on. Also, there is an unlimited number of each object type available. With this problem, the objective function and constraints remain the same, but the variable definitions now change because we can decide to take multiples of one type of item if we can maximize our total value and our constraints are satisfied. This can be illustrated by re-defining the integer variable

$$x_i \in \{0\} \cup \mathbb{N}$$

for each object $i \in \{1, \dots, 9\}$, where x_i is now the quantity of item i we will put in our knapsack. Solving this IP can be done through inputting the below code to LPSolve.

```
/* Objective function */
max: 72x_1+80x_2+66x_3+75x_4+68x_5+47x_6+85x_7+55x_8+82x_9;

/* Variable bounds */
c1: +10x_1+11x_2+7x_3+12x_4+3x_5+4x_6+9x_7+2x_8+8x_9 <= 40;
c2: +5x_1+6x_2+10x_3+4x_4+9x_5+8x_6+12x_7+10x_8+8x_9 <= 45;

/* Variables */
int x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9;
```

From the input, we can see that the input code is exactly the same with above except the last line. We see that the x_i variables are declared integers, not binary. This clarifies an important implication. The implication is that our additional constraint is sign restricted, also implied in the first problem.

$$x_i \geq 0$$

For all $i \in \{1, \ldots, 9\}$, x_i must be non-negative, meaning that we cannot have "negative" amount of any type of object, whatever that means.

The output we get this time is very interesting. We get a feasible solution, 7 improved solutions, the optimal solution, and a relaxed solution. Each solved maximum total value corresponds to,

solution	result
Relaxed	444.947916667
Feasible	335
Improved	387
Improved	410
Improved	420
Improved	428
Improved	430
Improved	433
Improved	441
Optimal	441

This is an IP so we cannot get the decimal relaxed solution. Then, the LPSolver finds a feasible solution, and then it continuously searches for improved solutions until it finds the optimal solution.

Variables	result
objective	441
x_1	0
x_2	1
x_3	0
x_4	1
x_5	3
x_6	0
x_7	0
x_8	0
x_9	1

The output tells us that the optimal max total value achievable with the constraints satisfied is 441. The optimal types of objects and their quantities we put in the knapsack are then one object 2, one object 4, three object 5s, and one object 9. (Note: from my Windows machine, a ThinkPad T470s, the entire process took 0.028 seconds, slightly longer than the 0.017 from solving with binary variables.) We can verify the constraints by plugging the types of objects and their optimal quantities back into the constraint. For the first constraint, summing the weights of the types of objects we take,

$$11 * 1 + 12 * 1 + 3 * 3 + 8 * 1 = 40 \leq 40$$

is satisfied. For the second, summing the volumes,

$$6 * 1 + 4 * 1 + 9 * 3 + 8 * 1 = 45 \leq 45$$

is also satisfied. Since the left and right side of both constraints are equal, after the optimal values of the decision variables are substituted back into the constraints, we can conclude that both constraints are binding. The constraint $x_i \geq 0$ is not all binding because we have non-zero solutions, the objects that we put in the knapsack. This also applies to the first part of the problem, which was implicit.