Optimization and Operations Research Assignment 2

Jingyu SUN

April 30, 2025

Question 1. Recall from the course notes that the knapsack problem involves choosing which items to put in a knapsack (or backpack). Each item has value, but you have a limited volume in the pack.

Consider the following linear program representing a knapsack problem for six items with total value z and maximum volume 35.

$$\begin{array}{lccccccccccccc} \max & z = & 19x_1 & + & 22x_2 & + & 30x_3 & + & 37x_4 & + & 11x_5 & + & 42x_6 & \\ & \text{s.t.} & 7x_1 & + & 6x_2 & + & 11x_3 & + & 13x_4 & + & 4x_5 & + & 13x_6 & \leqslant 35 \end{array}$$

with $x_i = 0, 1$, for $i = 1, \cdots, 6$.

(a) Relax the integral constraint to $x_i \geqslant 0$ (and real). Solve the relaxed version of the linear program either by inspection or with MATLAB.

(b) Relax the integral constraint to $0 \leqslant x_i \leqslant 1$ (and real). Solve the relaxed problem with MATLAB and submit via MATLAB Grader. You do not need to include your code.

(c) Now solve the true binary integer linear program with MATLAB and submit via MATLAB Grader. You do not need to include your code.

(d) State the solutions (variables and objectives) you found in 1(b) and 1(c). Comment on the relative size of the objective function for each solution from 1(a)-(c), and discuss why you would expect the value of the objective to change as it does.

(e) Now apply the greedy algorithm from the course notes to the problem. You should do this by hand, rather than in MATLAB, and include all working in your submission. How does the greedy solution compare to the optimal solution?

Solution 1. (a) For the first relaxed version, we will solve the question by hand to show the process.

To begin with, we will construct a tableau:

x_1	x_2	x_3	x_4	x_5	x_6	s_1	z	b
7	6	11	13	4	13	1	0	35
-19	-22	-30	-37	-11	-42	0	1	0

Then we will choose the pivot to be $(1, 1)$, which make the tableau like this:

x_1	x_2	x_3	x_4	x_5	x_6	s_1	z	b
1	$\frac{6}{7}$	$\frac{11}{7}$	$\frac{13}{7}$	$\frac{4}{7}$	$\frac{13}{7}$	$\frac{1}{7}$	0	5
0	$-\frac{40}{7}$	$-\frac{1}{7}$	$-\frac{12}{7}$	$-\frac{1}{7}$	$-\frac{47}{7}$	$\frac{19}{7}$	1	95

Still similar, we will choose the pivot to be $(1, 2)$, and the Simplex method stops since it has already found the optimal solution.

x_1	x_2	x_3	x_4	x_5	x_6	s_1	z	b
$\frac{7}{6}$	1	$\frac{11}{6}$	$\frac{13}{6}$	$\frac{2}{3}$	$\frac{13}{6}$	$\frac{1}{6}$	0	$\frac{35}{6}$
$\frac{20}{3}$	0	$\frac{31}{3}$	$\frac{32}{3}$	$\frac{11}{3}$	$\frac{17}{3}$	$\frac{11}{3}$	1	$\frac{385}{3}$

So here the optimal value for this relaxed version problem is $\frac{385}{3}$, and the optimal solution is $(0, \frac{35}{6}, 0, 0, 0, 0)$.

(d) Similarly, we find the optimal value for the second relaxed version and the actual integer linear program. So we have the following statements:

1. The optimal solution of the (b) is $(0, 1, 0, 1, 0.75, 1)$, and the optimal value is 109.25.

2. The optimal solution of the (c) is $(0, 1, 1, 0, 1, 1)$, and the optimal value is 105.

(e) To find the solution by using greedy algorithm, we will construct the table to calculate the ratio of each item, and find out the solution.

Items	x_1	x_2	x_3	x_4	x_5	x_6
Value	19	22	30	37	11	42
Weight	7	6	11	13	4	13
Ratio	2.714	3.667	2.727	2.846	2.750	3.231

And to present the steps we use to find the optimal solution, we will construct a tableau similar to Assignment 2 to show that.

Iterations	Current Weight	Current Value	Current Items
1	6	22	x_2
2	19	64	x_2, x_6
3	32	101	x_2, x_6, x_4
4	32	101	x_2, x_6, x_4
5	32	101	x_2, x_6, x_4
6	32	101	x_2, x_6, x_4

We find that from iteration 3, the weight would not be much higher since any items could not be less than the remaining weight.

So the optimal solution for the greedy algorithm is $(0, 1, 0, 1, 0, 1)$, and the value is 101.

Now compared with the previous solution by using MATLAB, we can see that the greedy algorithm is not optimal, this also indicates that the greedy algorithm would not always find the optimal solution.

Question 2. Apply a greedy heuristic from the course notes to find the shortest routes from the Haide College building (location 1) to the other 10 marked locations. Include all working in your submission.

As part of your submission include a table showing the shortest route to each destination and the length of the route. (Hint: A similar table is given in the solution for Tutorial 4, Question 3).

Solution 2. The greedy heuristic algorithm that applied in finding the shortest path is called Dijkstra's algorithm. We will also apply this by showing the iteration information. Below we ignore the unit and consider it as meters defaultly.

Iterations	1	2	3	4	5	6	7	8	9	10	11
Distance(1,1)	0	0	0	0	0	0	0	0	0	0	0
Distance(1,2)	∞	160	160	160	160	160	160	160	160	160	160
Distance(1,3)	∞	∞	560	560	560	560	560	560	560	560	560
Distance(1,4)	∞	∞	630	630	630	630	630	630	630	630	630
Distance(1,5)	∞	∞	510	510	510	510	510	510	510	510	510
Distance(1,6)	∞	590	590	580	580	580	580	580	580	580	580
Distance(1,7)	∞	∞	∞	∞	∞	620	620	620	620	620	620
Distance(1,8)	∞	∞	∞	∞	∞	∞	770	770	770	770	770
Distance(1,9)	∞	∞	∞	∞	∞	∞	∞	940	940	940	940
Distance(1,10)	∞	∞	∞	∞	∞	∞	∞	1030	1030	1030	1030
Distance(1,11)	∞	∞	∞	∞	∞	∞	910	910	910	910	910

Here we also exaplain each iteration's meanings: (To make the algorithm complete we also consider the path from location 1 to location 1 since it is not important here.)

Iteration 1: Initially we will start from location 1.

The distance from location 1 to location 2 is 160, which is less than ∞, so we will set it as 160.

The distance from location 1 to location 6 is 590, which is less than ∞, so we will set it as 590.

Iteration 2: From Iteration 1 we will select location 2 as the next location since the distance is the shortest.

The distance from location 1 to location 3 is 560, which is less than ∞, so we will set it as 560.

The distance from location 1 to location 5 is 510, which is less than ∞, so we will set it as 510.

Iteration 3: From Iteration 2 we will select location 5 as the next location since the distance is the shortest.

The distance from location 1 to location 4 is 630, which is less than ∞, so we will set it as 630.

The distance from location 1 to location 6 is 580, which is less than 590, so we will set it as 580.

Iteration 4: From Iteration 3 we will select location 3 as the next location since the distance is the shortest.

The distance from location 1 to location 4 is 840, which is higher than 630, so we would not change it.

Iteration 5: From Iteration 4 we will select location 6 as the next location since the distance is the shortest.

The distance from location 1 to location 7 is 620, which is less than ∞, so we will set it as 620.

Iteration 6: From Iteration 5 we will select location 7 as the next location since the distance is the shortest.

The distance from location 1 to location 8 is 770, which is less than ∞, so we will set it as 770.

The distance from location 1 to location 11 is 910, which is less than ∞, so we will set it as 910.

Iteration 7: From Iteration 6 we will select location 4 as the next location since the distance is the shortest.

The distance from location 1 to location 5 is already the shortest, so we would not change it.

The distance from location 1 to location 8 is 820, which is higher than 770, so we would not change it.

Iteration 8: From Iteration 7 we will select location 8 as the next location since the distance is the shortest.

The distance from location 1 to location 9 is 940, which is less than ∞, so we will set it as 770.

The distance from location 1 to location 10 is 1030, which is less than ∞, so we will set it as 910.

Iteration 9: From Iteration 8 we will select location 11 as the next location since the distance is the shortest.

There is no path from location 11 to other locations, so we would not change anything at this iteration.

Iteration 10: From Iteration 9 we will select location 9 as the next location since the distance is the shortest.

The distance from location 1 to location 10 is 1160, which is higher than 1030, so we would not change it.

The distance from location 1 to location 11 is already the shortest, so we would not change it.

Iteration 11: From Iteration 10 we will select location 10 as the next location since the distance is the shortest.

And then the algorithm stops since all the locations have been visited.

So we have the following table to show the shortest path from location 1 to other locations:

1. The shortest path from location 1 to location 1 is 0.
2. The shortest path from location 1 to location 2 is 160.
3. The shortest path from location 1 to location 3 is 560.
4. The shortest path from location 1 to location 4 is 630.
5. The shortest path from location 1 to location 5 is 510.
6. The shortest path from location 1 to location 6 is 580.
7. The shortest path from location 1 to location 7 is 620.
8. The shortest path from location 1 to location 8 is 770.
9. The shortest path from location 1 to location 9 is 940.
10. The shortest path from location 1 to location 10 is 1030.
11. The shortest path from location 1 to location 11 is 910.