IE 400 2019-2020 Fall Study Set 3 Solutions

(1) First calculate the ratios:

x_1	x_2	x_3	x_4	x_5
9/6	13/3	10/2	8/4	8/7

minimization so: $x_5 < x_1 < x_4 < x_2 < x_3$

The optimal solution of the LP relaxation: $x_{LP}^* = (1, 1, 2, \frac{3}{2}, 3)$

(1) $x^* = (1, 1, 2, \frac{3}{2}, 3)$ $z^* = 78$

$x_4 \leq 1$

$x^* = (1, \frac{5}{3}, 2, 1, 3)$ (2)
$z^* = 82,\bar{6}$
prune by bound

$x_4 \geq 2$

(3) $x^* = (\frac{2}{3}, 1, 2, 2, 3)$ $z^* = 79$

$x_1 \leq 0$

$x^* = (0, 1, 2, 2, 3)$ (4)
$z^* = 81$
prune by optimality

$x_1 \geq 1$

(5) $x^* = (1, 1, 2, 2, \frac{19}{7})$ $z^* = 79,71$

$x_5 \leq 2$

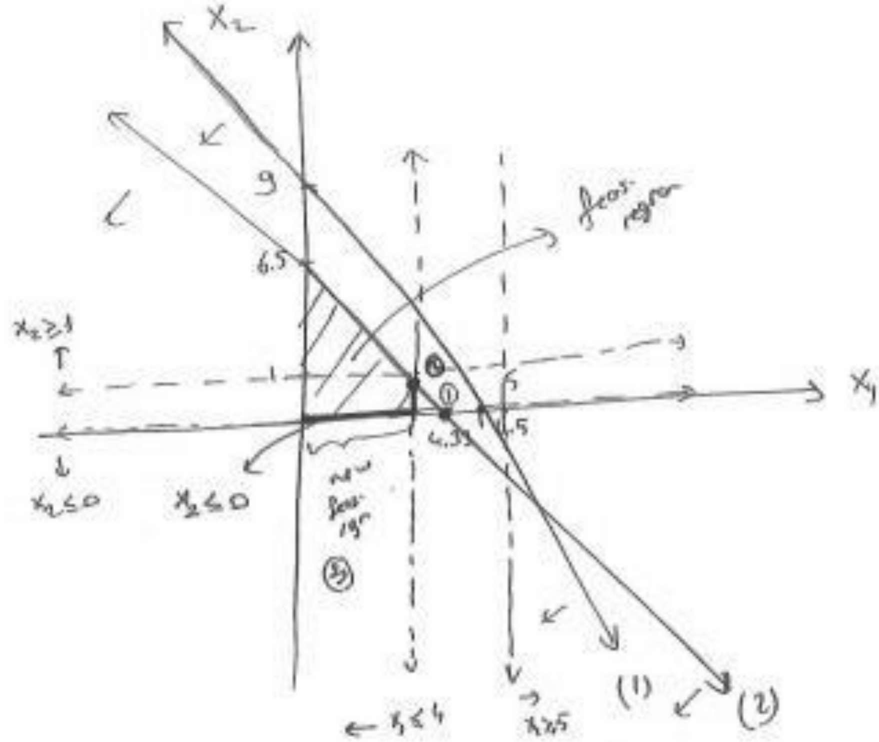
$x^* = (1, 1, 2, \frac{13}{4}, 2)$ (6)
$z^* = 84$
prune by bound

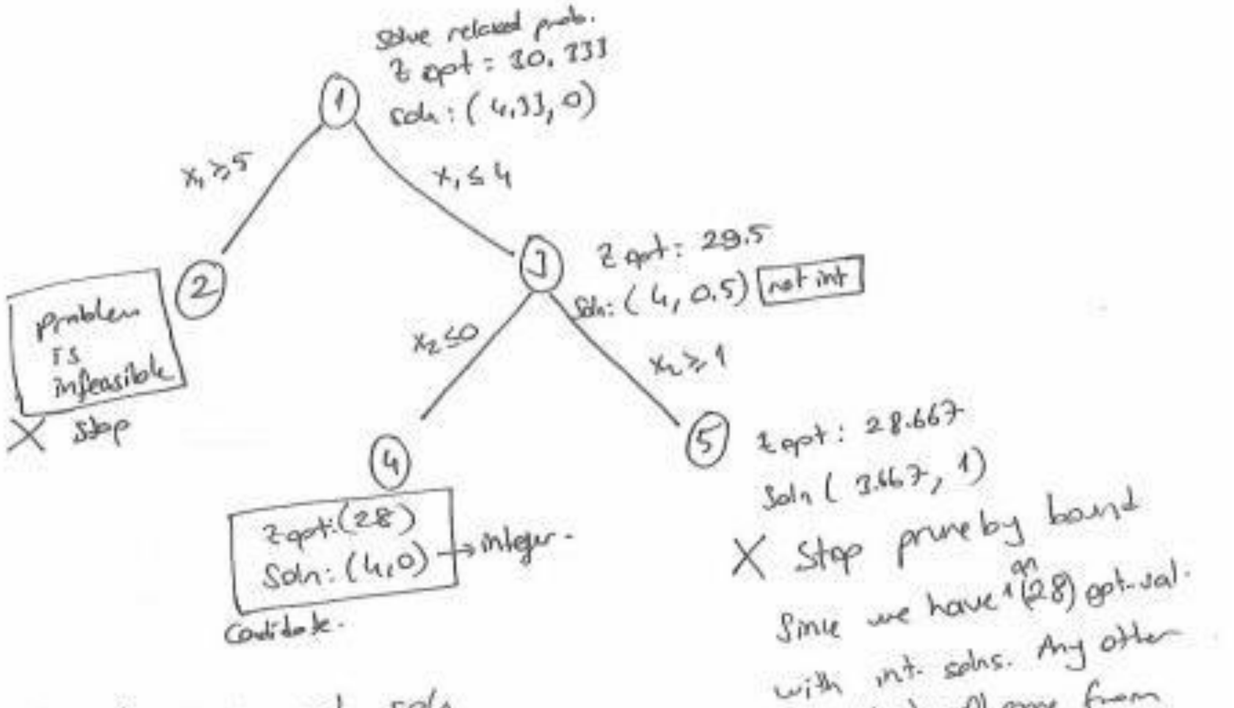
$x_5 \geq 3$

(7) $x^* = (1, 1, 2, 2, 3)$
$z^* = 82$
prune by bound

Optimal solution: $x_{IP}^* = (0, 1, 2, 2, 3)$ $z_{IP}^* = 81$

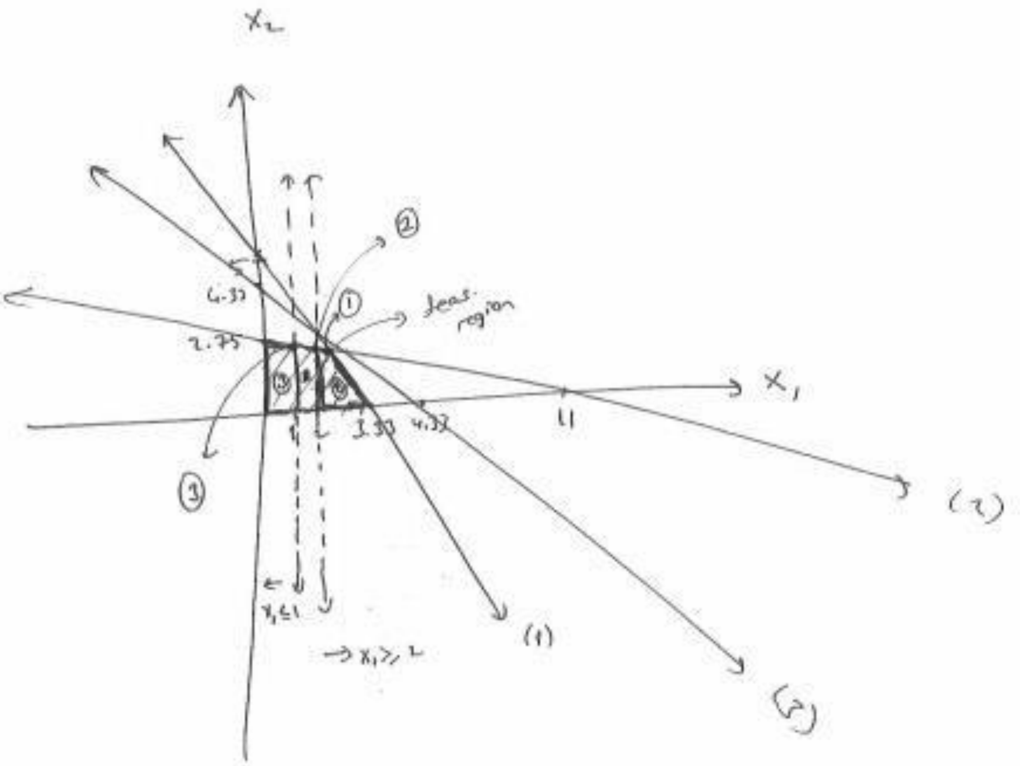
2)



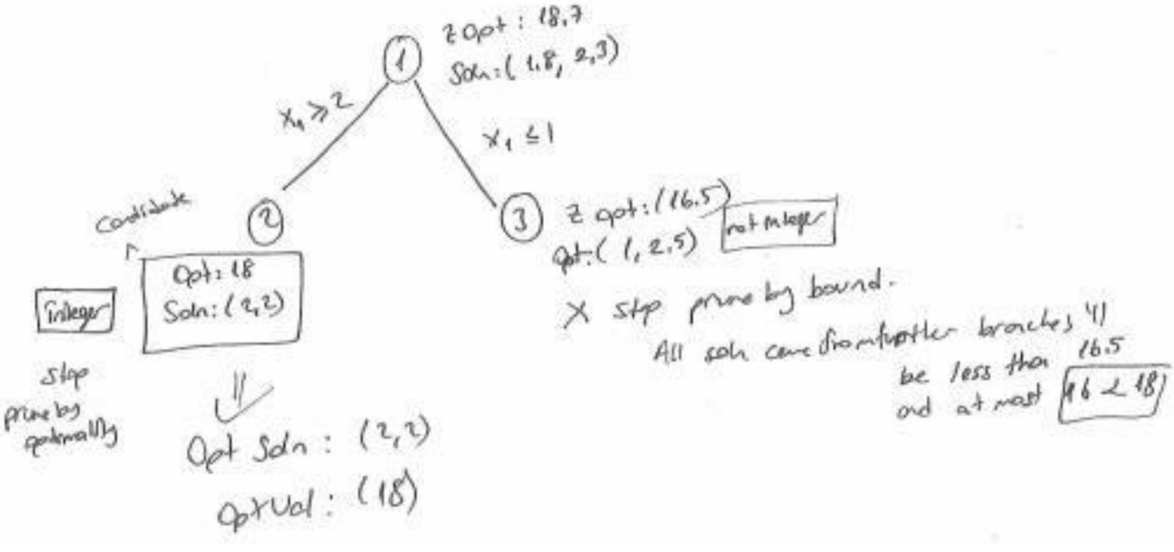


So (4,0) is an opt soln
Opt Value is 28.

3)



x2
x1
4.33
2.75
3.33
4.33
11
feas. region
x1 ≤ 1
→ x1 ≥ 2
(1)
(2)



z opt : 18.7
Soln: (1.8, 2.3)
x1 ≥ 2
x1 ≤ 1
candidate
Opt: 18
Soln: (2,2)
integer
stop
prune by optimality
z opt: (16.5)
Opt: (1, 2.5)
not integer
X stop prune by bound.
All soln come from further branches will be less than 16.5
and at most 16 < 18
Opt Soln : (2,2)
Opt Val : (18)

4) 1-2-5 with length 14

5)

This problem should be formulated as a shortest path problem.

The network will have six nodes.

- Node i is the beginning of year i and for $i<j$, an arc (i,j) corresponds to purchasing a new car at the beginning of year i and keeping it until the beginning of year j.

The length of arc (i,j) (call it c_{ij}) is the total net cost incurred from year i to j.

c_{ij} = maintenance cost incurred during years $i, i+1, \ldots, j-1$
+ cost of purchasing a car at the beginning of year i
- trade-in value received at the beginning of year j

Applying this formula to the information the problem yields

C12=300+12000-7000=5300

C13=300+500+12000-6000=6800

C14=300+500+800+12000-4000=9600

C15=300+500+800+1200+12000-3000=11800

C16=300+500+800+1200+1600+12000-2000=13500

C17=300+500+800+1200+1600+2200+12000-1000=16700

C23=300+12000-7000=5300

C24=300+500+12000-6000=6800

C25=300+500+800+12000-4000=9600

C26=300+500+800+1200+12000-3000=11800

C27=300+500+800+1200+1600+12000-2000=13500

C34=300+12000-7000=5300

C35=300+500+12000-6000=6800

C36=300+500+800+12000-4000=9600

C37=300+500+800+1200+12000-3000=11800

C45=300+12000-7000=5300

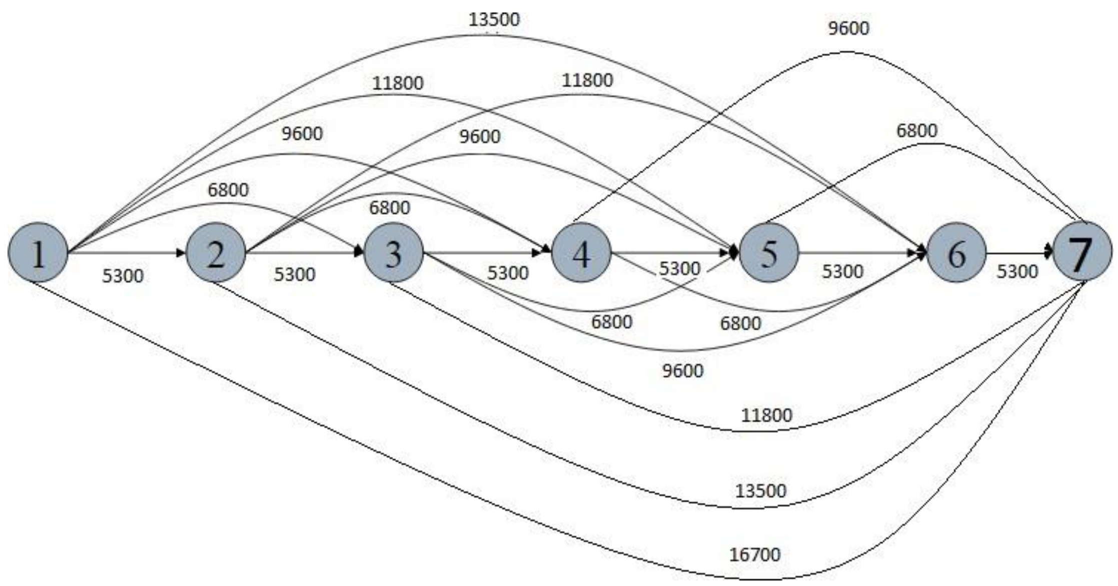
C46=300+500+12000-6000=6800

C47=300+500+800+12000-4000=9600

C56=300+12000-7000=5300

C57=300+500+12000-6000=6800

C67=300+12000-7000=5300



From the figure above, we can conclude that replacing car at years 2,4 and 6 yields the least costly solution.

6)

b) Step 0: $P=\{1\}$ $T=\{2,\dots,9\}$

$V[1]=0$

$V[2]=18$ $V[3]=22$ $V[4]=20$ $V[5]=30$

$V[6]=V[7]=\dots=V[9]=+\infty$

Step 1: Choose $k=2$

Step 2: $P=\{1,2\}$ $T=\{3,\dots,9\}$

$V[2]=18$

$V[3]=\min\{22,33\}=22$ $d[2]\leftarrow 1$

$V[6]=18+27=45$ $d[6]\leftarrow 2$

$V[8]=18+33=51$ $d[8]\leftarrow 2$

Step 1: Choose $k=4$

Step 2: $P=\{1,2,4\}$ $T=\{3,5,6,7,8,9\}$

$V[4]=20$ $d[4]\leftarrow 1$

$V[6]=\min\{20+16,45\}=36$ $d[6]\leftarrow 4$

$V[7]=20+30=50$ $d[7]\leftarrow 4$

$V[5]=\min\{30,51\}=30$

$V[3]=\min\{33,22\}=22$

Step 1: $k=3$

Step 2: $V[3]=22$

$V[6]=\min\{36,22+45\}=36$

Step 1: $k=5$ $P=\{1,2,4,5\}$ $T=\{3,6,7,8,9\}$

Step 2: $V[5]=30$ $d[5]\leftarrow 1$

$V[9]=30+28=58$ $d[9]\leftarrow 5$

$V[6]=\min\{30+19,36\}=36$

$V[7]=\min\{30+22,50\}=50$

Step 1: $k=6$

Step 2: $P=\{1,\dots,6\}$ $T=\{7,8,9\}$

$V[6]=36$

$V[7]=\min\{36+29,50\}=50$ No updates.

$V[9]=\min\{36+35,58\}=58$

$V[8]=\min\{36+32,51\}=51$

Step 1: $k=7$

Step 2: $P=\{1,\dots,7\}$ $T=\{8,9\}$

$V[7]=50$

$V[9]=\min\{50+25,58\}=58$ No updates

Step 1: $k=8$

Step 2: $P=\{1,\dots,8\}$ $T=\{9\}$

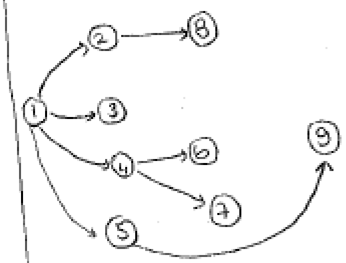
$V[8]=51$

$V[9]=\min\{51+34,58\}=58$ No update

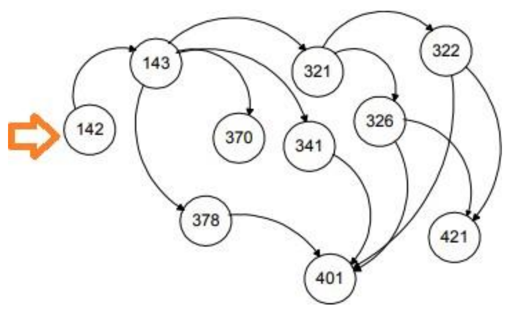
Step 1: $k=9$

Step 2: $P=\{1,\dots,9\}$ $T=\{\}$

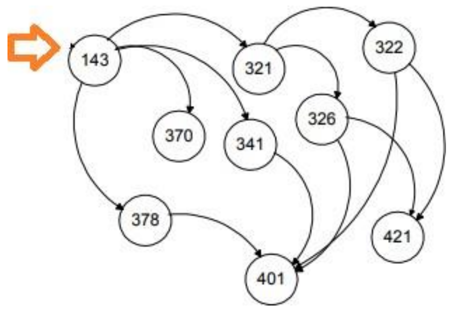
Shortest path tree



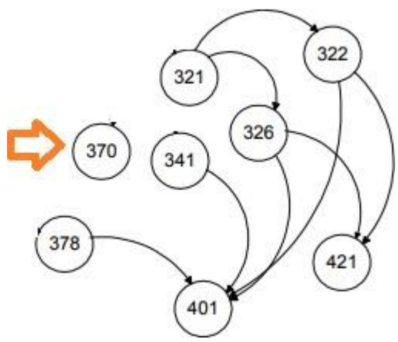
7) Pick the node without any incoming arcs:



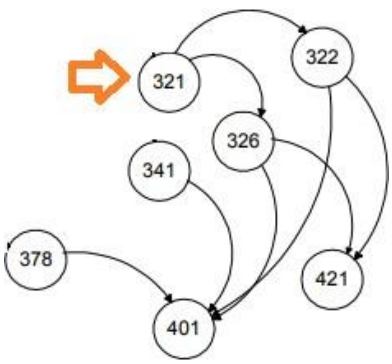
Randomly select another node without any incoming arcs and repeat:



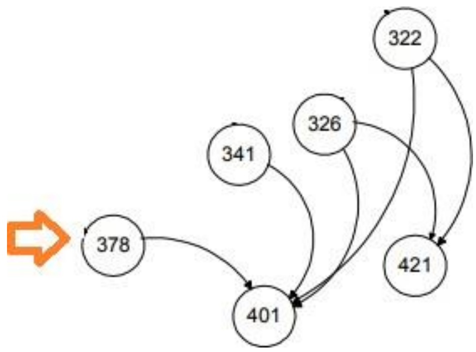
142-143



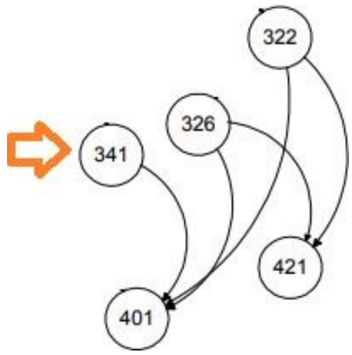
142-143-370



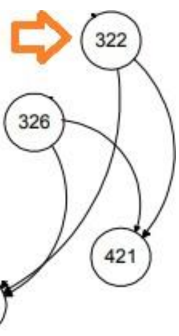
142-143-370-321



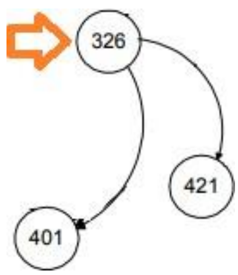
142-143-370-321-378



142-143-370-321-378-341



142-143-370-321-378-341-322



142-143-370-321-378-341-322-401-422

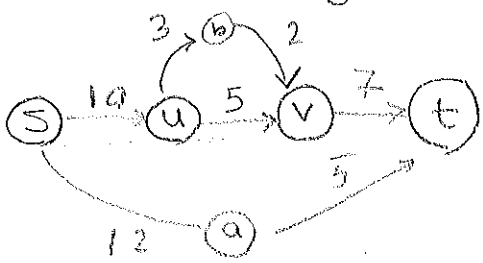
(8)

t	$v^{(t)}(A)$	$v^{(t)}(B)$	$v^{(t)}(C)$	$v^{(t)}(D)$	$v^{(t)}(E)$	d(A)	d(B)	d(C)	d(D)	d(E)
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$					
1	0	-1	4	$+\infty$	$+\infty$		A	A		
2	0	-1	2	1	1			B	B	B
3	0	-1	2	-2	1				E	

remember that $v^{(t)}[K] = \min\limits_{(i,K)} \{ v^{t-1}[i] + c_{iK} \}$

(9) a) No, principle of optimality doesn't apply.

Counterexample



Shortest path from S. to t $\begin{cases} S-U-V-t = 10 \\ \text{or} \\ S-U-b-v-t = 10 \end{cases}$ but Shortest path from u to v $\{ U-b-V = 3$

b) $V[S] = 0$

$$V[i] = \min_{(i,K)\in A} \{ \text{Max} \{ V[K],\ c_{iK} \} \}$$

c) you can adapt dijkstra algorithm with the functional equations stated in part (b)

10)

a) False. Consider nodes A,B,C, cost of edge (A,B) is 3, edge (A,C) is 1 and (B,C) is 2. There are 2 shortest paths from A to B both of length 3.

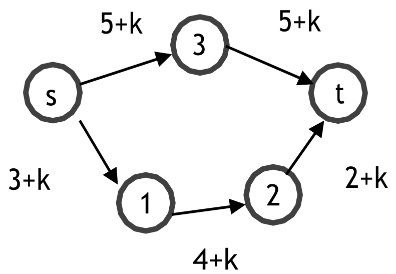
b) True. Both algorithms are guaranteed to produce the same shortest path weight, but if there are multiple shortest paths, Dijkstra's will choose the shortest path according to the greedy strategy, and Bellman-Ford will choose the shortest path depending on the order of relaxations, and the two shortest path trees may be different.

c) False. For example a graph G = (V, E) = ({a, b, c}, {(a, b), (a, c)}) has valid topological orderings [a, b, c] or [a, c, b].

d) False. Undirected graph on nodes A,B,C. Edges (A,B) and (B,C) have cost 1 and edge (A,C) has cost 3. Length of shortest A to C path is 2. If each edge cost is increased by k = 10 the shortest path length becomes 13. But 13 − 2 is not a multiple of 10.

e) False. Consider graph with nodes A,B,C, arcs (B,A), (A,C) and (C,B) have cost 1. In the directed graph, shortest distance from A to B is 1 + 1 = 2, if it becomes undirected, the shortest distance becomes 1.

f) False



without adding k, the shortest path is s-1-2-t with objective function of 9 but if k>1 then the shortest path changes to s-3-t.