HW Solutions #4

ELEN E4710 - Intro to Network Engineering — Due 11/15/2004
Fall 2004 — Prof. Rubenstein

Homework must be turned in at the beginning of class on the due date indicated above. CVN students have one additional day. Late assignments will not be accepted.

1. Let G be a graph where each edge has 2 weight functions, w and v, where $v(e) = w(e) + 1$, $w(e) > 0$ for each edge $e \in G$.

   (a) If tree T is a minimum spanning tree using weight function w, is it also a minimum spanning tree using weight function v? Explain why this is true or give a counterexample.

   Ans)

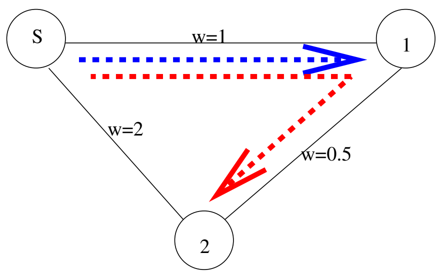
   Yes, it is.

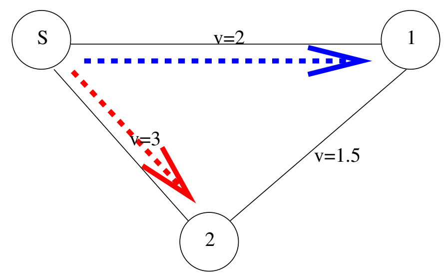
   Because the number of edges of any tree equals $N - 1$, where N is the number of nodes. Suppose solution S minimizes $\sum_{e \in S} w(e)$, it also minimizes $\sum_{e \in S} v(e) = \sum_{e \in S}(w(e) + 1) = \sum_{e \in S} v(e) + (N - 1)$. ■

   (b) If tree T is a shortest path tree using weight function w, is it also a shortest path tree using weight function v? Explain why this is true or give a counterexample.

   Ans)

   No, it isn't.





   A counter example is the above. The shortest path tree for w is $\{S \to 1 \to 2\}$. But the shortest path tree for v is $\{S \to 1, S \to 2\}$. ■

2. Compute an MST for the above graph using either algorithm covered in class.

   Ans)

   Using Kruskal algorithm, we can take the following steps:
   {AB}
   {AB,CD}
   {AB,CD,FG}
   {AB,CD,FG,BE}
   {AB,CD,FG,BE,BF}
   {AB,CD,FG,BE,BF,DG}

   Finally, {AB,CD,FG,BE,BF,DG} is the minimum spanning tree. ■

3. Compute a shortest path tree for the above graph rooted at node A.

   (a) using Dijkstra's algorithm.

   Ans)

{AB}
{AB,BE}
{AB,BE,BF}
{AB,BE,BF,FG}
{AB,BE,BF,FG,GD}
{AB,BE,BF,FG,GD,DC}

∎

(b) using the Bellman-Ford algorithm (include the information about the predecessor node).

Ans)

The Bellman-Ford matrix is the following:

	B	C	D	E	F	G
0	∞	∞	∞	∞	∞	∞
1	1,A	7,A	∞	5,A	∞	∞
2	1,A	7,A	8,C	2,B	3,B	8,E
3	1,A	7,A	8,C	2,B	3,B	4,F
4	1,A	7,A	6,G	2,B	3,B	4,F

∎

(c) Assume that after the Bellman-Ford algorithm completes (i.e., no further changes are made to the tree), the weight of edge (E, D) changes to 1. Continue the algorithm to find the new shortest path.

Ans)

The new Bellman-Ford matrix is the following:

	B	C	D	E	F	G
Previous	1,A	7,A	8,C	2,B	3,B	4,F
1	1,A	7,A	3,E	2,B	3,B	4,F
2	1,A	4,D	3,E	2,B	3,B	4,F

∎

(d) Assume that after the Bellman-Ford algorithm completes for a second time, edge (F, G) changes its weight to 4. Continue the algorithm once more to find the new shortest path.

Ans)

The new Bellman-Ford matrix is the following:

	B	C	D	E	F	G
Previous	1,A	4,D	3,E	2,B	3,B	4,F
1	1,A	4,D	3,E	2,B	3,B	5,D

∎

4. Consider the small 4-node network above with weighted edges. Assume that the nodes communicate on a round-by-round basis, where a node can communicate once to each of its neighbors within a single round, but only one node can send a message in a given round.

   (a) Run the Bellman-Ford Algorithm (i.e., Distance Vector Protocol) without Poison Reverse/Split Horizon implemented to compute the shortest paths trees to all nodes.

   Ans)

	B	C	D
0	∞	∞	∞
1	1,A	∞	∞
2	1,A	∞	2,B
3	1,A	3,D	2,B

∎

(b) Assume that the weight of edge (A, B) suddenly changes to 200. Perform the first five rounds of the algorithm after the change. Indicate how many rounds take place in total until convergence.

Ans)

	B	C	D
Previous	1,A	3,D	2,B
1	3,D	3,D	2,B
2	3,D	3,D	4,B
(3-1)/2*3=3	3,D	5,D	4,B
4	5,D	5,D	4,B
5	5,D	5,D	6,B
(5-1)/2*3=6	5,D	7,D	6,B
	$\vdots$	$\vdots$	$\vdots$
(99-1)/2*3=147	99,D	100,A	100,B
148	101,D	100,A	100,B
149	101,D	100,A	101,C
150	102,D	100,A	101,C

From the Bellman-Ford matrix, we need 150 rounds to converge. ∎

(c) Run the Bellman-Ford Algorithm on the original graph (where (A, B) has weight 1) with Poison Reverse/Split Horizon implemented to compute the shortest paths trees to all nodes.

Ans)

	B	C	D
0	∞	∞	∞
1	1,A	∞	∞
2	1,A	∞	2,B
3	1,A	3,D	2,B

∎

(d) Assume again that the weight of edge (A, B) suddenly changes to 200. Perform all rounds of the algorithm until messages cease to be sent.

Ans)

	B	C	D
Previous	1,A	3,D	2,B
1	200,A	3,D	2,B
2	200,A	3,D	201,B
3	200,A	100,A	201,B
4	200,A	100,A	101,C
5	102,D	100,A	101,C

∎

(e) Assume an additional edge (B, C) with weight 3 is added to the original graph. Modify the routing tables from part 4c to include this edge when Poison Reverse/Split Horizon is implemented (you need not draw all the steps, just the final tables).

Ans)

	B	C	D
0	∞	∞	∞
1	1,A	∞	∞
2	1,A	∞	2,B
3	1,A	3,D	2,B

∎

(f) Suppose the weight of (A, B) once again shifts to 200. Perform the first 5 iterations of the algorithm. Explain why Poison Reverse did not prevent the count to infinity problem.

Ans)

	B	C	D
Previous	1,A	3,D	2,B
1	6,C	3,D	2,B
2	6,C	7,B	2,B
3	6,C	7,B	8,D
4	11,C	7,B	8,D
5	11,C	12,B	8,D

∎

(g) Did Poison Reverse help at all in part 4f? Explain your answer.

Ans)

Poison reverse helps, in that it takes less time to converge, but does not eliminate the count-to-infinity problem. Because it cannot detect the circle $B \to C \to D \to B$. These three nodes compute the shortest path length depending on each other. ∎