Problem Set 5

Noella James

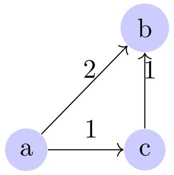
03/17/2017

collaborators: none

Problem 5-3: Different paths

Solution

If we are given a Graph $G = (V, E)$ with nonnegative edge weights, the shortest path found by Dijkstra will differ from the shortest path from Bellman-Ford. Take, for example, the following counterexample:



The length of each edge is of size 1. If we had to find the shortest path from vertex a to vertex b, Dijkstra and Bellman-Ford will give two different paths. The order of E be will $(a, c), (c, b), (a, b)$.

Dijkstra

Following Dijkstra's algorithm, the path chosen will be as follows

$$\begin{bmatrix} a & b & c \\ 0 & \infty & \infty \\ - & 2 & 1 \\ - & 2 & - \end{bmatrix}$$

Path chosen from vertex a to vertex b: Edge (a, b)

When running through Dijkstra's algorithm, we come to the conclusion that even though there is an order E to follow and pick edges, Dijkstra's shortest path from vertex a to vertex b will consist of edge (a, b) and not edges (a, c) and (c, b) since Dijkstra's algorithm ends up evaluating edge (a, b) before edge (c, b). The total cost of (a, c) and (c, b) is 2, which is exactly the same as the cost of (a, b). Therefore, since Dijkstra will only update the path if it is new cost is less than the current cost, the shortest path remains as (a, b).

Bellman-Ford

Following Bellman-Ford's algorithm, the path chosen will be as follows

$$\begin{bmatrix} b & 0 \\ c & 1 \\ a & 2 \end{bmatrix}$$

Path chosen from vertex a to vertex b: Edges $(a, c), (c, b)$

While running through Bellman-Ford's algorithm, we come to the conclusion that Bellman-Ford is more impacted by the edge order E. Bellman-Ford's shortest path from vertex a to vertex b will consist of edges (a, c) and (c, b) and not edge (a, b). This is because Bellman-Ford's algorithm iterates through all possible edges in G (hence the importance of edge order E). Dijkstra's, on the other hand, iterates through all the vertices and then looks at neighbors of those vertices. Thus, Bellman-Ford will evaluate edges (a, c) and (c, b) prior to edge (a, b). Since, as we determined before, the costs between these two paths are the same, Bellman-Ford will not update it's current shortest path.

Conclusion

Depending on the edge order E, if we are given a Graph $G = (V, E)$ with nonnegative edge weights, Dijkstra's algorithm and Bellman-Ford's algorithm will provide two different shortest paths. Although both are correct, the shortest paths found by Bellman-Ford are not the same as the ones found by Dijkstra's algorithm because of subtle differences within the algorithms. Both paths will have the same total cost. However, this will only work on non-distinct edge weights (2 or more edges have the same weight). If the edge weights are distinct, both algorithms will provide the same shortest path.