Camilla Bocciolone

Alba Ballarà Sala

Assignment 4

ex 4

In this experiment we had to compare Dijkstra's and Bellman-Ford's algorithms performance on random graphs. In order to do that we decided to use the number of times we relaxed an edge in the 2 algorithms, by using an attribute called iteration that was initialized to 0 and it was incremented every time we did an edge relaxation.

Since we could assume the required conditions for each algorithm we tested the algorithms on 10 random graphs of 1000 vertices with only positive weight. To implement these graphs we used the method generateRandomGraph2 that chose randomly for each vertex the number of edges and their destination.

By comparing the number of times the edges were relaxed in order to find the single source shortest paths using the 2 algorithms we got the following output

To find the shortest path using Dijkstra in graph 0 I had to relax 4613 on a total of 4653 edges
To find the shortest path using Bellman in graph 0 I had to relax 7596 on a total of 4653 edges

To find the shortest path using Dijkstra in graph 1 I had to relax 4360 on a total of 4426 edges
To find the shortest path using Bellman in graph 1 I had to relax 7884 on a total of 4426 edges

To find the shortest path using Dijkstra in graph 2 I had to relax 4487 on a total of 4548 edges
To find the shortest path using Bellman in graph 2 I had to relax 7814 on a total of 4548 edges

To find the shortest path using Dijkstra in graph 3 I had to relax 4381 on a total of 4476 edges
To find the shortest path using Bellman in graph 3 I had to relax 7326 on a total of 4476 edges

To find the shortest path using Dijkstra in graph 4 I had to relax 0 on a total of 4302 edges
To find the shortest path using Bellman in graph 4 I had to relax 0 on a total of 4302 edges

To find the shortest path using Dijkstra in graph 5 I had to relax 4555 on a total of 4563 edges
To find the shortest path using Bellman in graph 5 I had to relax 7795 on a total of 4563 edges

To find the shortest path using Dijkstra in graph 6 I had to relax 4205 on a total of 4301 edges
To find the shortest path using Bellman in graph 6 I had to relax 7213 on a total of 4301 edges

To find the shortest path using Dijkstra in graph 7 I had to relax 4365 on a total of 4431 edges
To find the shortest path using Bellman in graph 7 I had to relax 6873 on a total of 4431 edges

To find the shortest path using Dijkstra in graph 8 I had to relax 4364 on a total of 4428 edges
To find the shortest path using Bellman in graph 8 I had to relax 7868 on a total of 4428 edges

To find the shortest path using Dijkstra in graph 9 I had to relax 4441 on a total of 4514 edges

To find the shortest path using Bellman in graph 9 I had to relax 8064 on a total of 4514 edges

Our expectation were that: because of its greedy strategy in Dijkstra's algorithm each edge would have been relaxed at most once during the entire process while in Bellman-Ford because of the necessity to handle graphs with negative edge weights each edge would have been relaxed multiple times. In each iteration of the Bellman-Ford algorithm a vertex is in fact dequeued and all adjacent edges are relaxed leading to multiple relations of the same edge.

The results we found met our expectations since the number of relaxations made in Dijkstra are always smaller than the number of relaxation made by BellmanFord algorithm. We can see that the number of relaxation in Dijkstra is smaller than the number of edges and since we set the max number of edges to (numVertices)/100=1000/100=10, our graph is quite sparse and it is difficult that all edges are relaxed. The number of relaxation in Bellman Ford is instead bigger than the number of edges. Finally we can see that sometimes there is no relaxation from the source (for example in graph 4) because the random number of edges from the source can be 0 and in that case it is detected by both the algorithms that count 0 edge relaxation.

From our expectation and result we can conclude that for graphs with positive weight Dijkstra performs better because it requires less edge relaxation.