ShortestCycle

Isaac Gutt

February 10, 2022

1 The Algorithm

To solve the shortest cycle problem, I used Dijkstra's shortest paths algorithm. I removed the edge of interest from the graph, adding it to a list x, then ran Dijkstra's algorithm from one vertex of the edge to the other, receiving that path and adding it to list x. This results in a list with the shortest cycle in the graph containing the edge of interest. I used Sedgewick's DijkstraUndirectedSP.java, IndexMinPQ.java, EdgeWeightedGraph.java, Bag.java.

2 Prove the correctness of your algorithm

Let v be the source vertex
Let $d(v, w)$ be the shortest path found by the algorithm and let $\delta(v, w)$ be the shortest cycle e containing e = $v \to w$, the edge of interest
Let G^1 be the graph
Lemma: Dijkstra's algorithm correctly results in $d(v, w)$

Theorem 1. *If we remove the edge of interest from the graph, call Dijkstra's algorithm from v to w, and add the edge of interest to that path, we end up with the shortest cycle containing the edge of interest.*
Math: In $G^1 - e$, $d(v, w) + e = \delta(v, w)$

Definition: a cycle is a trail in which only the first and last vertices are equal.

By having a path $p = d(v, w)$ after calling Dijkstra's algorithm, adding the edge of interest e from v to w (or w to v since undirected) to p will create a cycle because path will start and end with v or w. This must be the shortest cycle because we already have the shortest path p, and the solution requires edge e in the path.

Proof. Since Dijsktra's algorithm is proven above to find the shortest path between two vertices, adding the edge of interest e to the shortest path p will result in $\delta(x)$. $\square$

3 Prove the Algorithm's order of growth

Using a min-priotity queue, Dijkstra's algorithm has $O((E + V)logV)$ performance.
My solution only uses one call of Dijkstra's algorithm to find the solution. Therefore, it's order of growth is also $O((E + V)logV)$.