A Linear Time Algorithm for the Longest Path Problem on Cactus Graphs

Minko Markov*

October 16, 2009

Abstract

To compute the length of a path of maximum length in an undirected graph is a problem that arises naturally. The graph can have positive edge weights, in which case the length of any path is the sum of the weights of its edges, or not, in which case the length of a path is the number of edges in it. Both versions are known to be $\mathcal{NP}$-complete.

A way to tackle with $\mathcal{NP}$-completeness on graphs is to construct fast, *i.e.* polynomial time, algorithms on restricted graph classes. A linear time algorithm for Longest Path on edge weighted trees was constructed by Dijkstra around 1960. For several decades the trees were the only natural graph class for which a polynomial time algorithm for Longest Path was known. Then Uehara and Uno [1] proposed polynomial time algorithms for that problem on cacti and block graphs. The said two algorithms are based on generalization of Dijkstra's algorithm for trees. Their time complexities are $O(|V(G)|^2)$ and $O(|V(G)| + |E(G)|)$, respectively.

We propose an algorithm with $O(|V(G)|)$ time complexity to compute the length of a longest path in cactus graphs. Our algorithm is not based on Dijkstra's algorithm for trees but on a radically different idea. First turn the cactus into a rooted cactus, which can be done in linear time, and then work from the leaves upwards to the root. Each vertex u of the rooted cactus is associated with two numbers: the length of a longest path and the length of a longest path having u as one endpoint, in the subcactus rooted at u. We call that ordered pair, *the label* of the vertex. Clearly, the desired answer is the first number in the label of the root. The label of each nonleaf vertex is computed only from the labels of its children.

Our algorithm on cacti uses as an auxilliary algorithm FC that, given a cycle s and the labels of its vertices, computes the length of a longest path that contains at least one edge of the cycle in time $\Theta(|s|)$. It turns out FC can accomplish that even on graphs that are more general than cacti under the following two provisions. First, the cycle is chordless and the removal of its edges leads to the appearance of $|s|$ connected components, and second, for each such component we know the length of a longest path in it with one endpoint the corresponding cycle vertex.

References

[1] Ryuhei Uehara and Yushi Uno. On computing longest paths in small graph classes. *International Journal of Foundations of Computer Science*, 18(5):911–930, 2007.

*Department of Computing Systems, Faculty of Mathematics and Informatics, "St. Kliment Ohridski" University of Sofia, 5 J. Bourchier Blvd, P.O. Box 48, BG-1164 Sofia, Bulgaria. Email: minkom@fmi.uni-sofia.bg