International Workshop On
Combinatorial Algorithms
IWOCA

http://www.iwoca.org—Problems section.

LONGEST PATH with Neighborly Help under Edge Deletion

Fabian Frei
ETH Zurich
fabian.frei@inf.ethz.ch

presented at IWOCA 2020
10-06-2020

In the NP-hard optimization problem LONGEST PATH, we are given a graph (a simple graph, i.e., undirected, unweighted, with neither loops nor parallel edges) and asked to find a longest path (a simple path, i.e., without repeating verticies).

We consider LONGEST PATH in the recently introduced *Neighborly Help* model [1], which allows us to query an oracle for optimal solutions to *neighbor instances*. The notion of a neighbor needs to be specified for every problem. We examine the local modification of an edge deletion: We consider as neighbors exactly those graphs that result by removing from the given graph a single edge.

Given a graph, we may therefore query the oracle for a longest path in any one-edge-deleted subgraph of our choice.

Three Queries: In P. With three queries, the problem can be solved in polynomial time as follows: We pick an arbitrary vertex that is incident to at least three edges. (If there is no such vertex, the instance is trivial anyway.) We then query the oracle an optimal solution to the three neighbors that result from deleting three of the incident edges. At least one of these three neighbor graphs has a longest path that is also a longest path for the given instance. This is because a longest path in the given graph remains valid unless one of the path edges is deleted, and every vertex can be incident to at most two edges of the (simple) path. One of the three optimal neighbor solutions is thus optimal for the given graph as well.

One Query: NP-Hard. If we restrict ourselves to a single query, however, then LONGEST PATH remains NP-hard by the following Turing reduction [1, Theorem 17]. Assuming there was an optimal polynomial-time algorithm for LONGEST PATH that uses only a single edge-deletion query, we could also find a longest path in polynomial time with no queries at all: Given any graph, we first delete all edges, turning it into an independent set with the trivial longest path. We then re-insert the edges one by one while maintaining a longest path for all intermediate graphs by using the assumed algorithm and simulating the oracle ourselves.

Two Queries: Open. It remains open whether the problem is still NP-hard for two queries.

References

[1] Elisabet Burjons, Fabian Frei, Edith Hemaspaandra, Dennis Komm, and David Wehner. Finding Optimal Solutions With Neighborly Help. In *Proc. MFCS*, volume 1382 of *LIPIcs*, pages 78:1–78:14, 2019. The full version including the appendices is available at `https://arxiv.org/abs/1906.10078`.