Introduction and Motivation

The all pairs shortest path (APSP) problem is one of the most fundamental problems in theoretical computer science. The trivial classical algorithm for APSP and most all pairs path problems runs in \(O(n^3)\) time, while the trivial algorithm in the quantum setting runs in \(O(n^{1.5} \cdot \log n)\) time. A major open problem in classical algorithms is to obtain a truly subcubic time algorithm. A fast quantum algorithm in the quantum setting runs in \(O(n^{1.25} \cdot \log n)\) time. The all pairs shortest path (APSP) problem is one of the most fundamental examples of such problems. A simple classical algorithm for the APSP problem is to compute the distances between vertices of a graph, and then to find a shortest path for each pair of nodes with a fast quantum counterpart.

Quantum Algorithms for Shortest Paths Problems in Structured Instances

Aran Nayebi

Introduction and Motivation

The all pairs shortest path (APSP) problem is one of the most fundamental problems in theoretical computer science. The trivial classical algorithm for APSP and most all pairs path problems runs in \(O(n^3)\) time, while the trivial algorithm in the quantum setting runs in \(O(n^{1.5} \cdot \log n)\) time. A major open problem in classical algorithms is to obtain a truly subcubic time algorithm for APSP, i.e. an algorithm running in \(O(n^{2+\epsilon})\) time for constant \(\epsilon > 0\). To approach this problem, many truly subcubic time classical algorithms have been devised for APSP and its variants for structured instances. Some examples of such problems are APSP in geometrically weighted graphs, graphs with small integer edge weights or a small number of weights incident to each vertex, and the all pairs earliest arrivals problem (which has applications to traffic light scheduling). A natural analogue to this question is whether APSP has a nontrivial (i.e. \(O(n^{2+\epsilon})\) time for \(\epsilon > 0\)) quantum algorithm.

We revisit these aforementioned structured instances in the quantum setting and exhibit the first nontrivial (i.e. \(O(n^{2+\epsilon})\) time) quantum algorithms for the problems.

Quantum Preliminaries

- **Quantum algorithm**: A non-deterministic algorithm that access the entry of any input matrix in a random access manner, which is standard (ref. [1] [21]).
- **Given an n × n matrix A, we have an oracle \(O_A\) such that for any \(i, j \in [n]\), \(A[i][j] = |\psi_{ij}\rangle\langle\psi_{ij}|\).
- **We are interested in time complexity (and not simply query complexity)**, so we count all the computational steps of the quantum algorithms we give and assign unit cost for each call to the oracle \(O_A\).
- **High probability**: With regard to an algorithm outputting a desired result means with probability at least \(2/3\). This probability can be boosted in a standard way to \(1 - 1/poly(n)\).

Summary of Results

We take \(n < 2^{37} \approx 2.7 \cdot 10^{11}\). The best-known classical approaches, which are standard (ref. [1] [21]), have a time complexity of \(O(n^{1.3})\) for the geometrically weighted graph case, \(O(n^{1.5})\) for the Euclidean graph case, and \(O(n^{2+\epsilon})\) for the all pairs shortest path problem in geometrically weighted graphs.

Quantum Tools and Methods

- **The main idea behind our techniques is to examine the classical algorithms for the problems and replace key parts of the algorithms with a faster quantum counterpart**.
- **All algorithms for APSP use the following two ingredients in one way or another**.
  1. Define a matrix product over some algebraic structure and iterate this matrix product \(t\) times to compute the distances between pairs of vertices that have shortest paths on few vertices. For parameter \(t\) nodes.
  2. Pick a random sample \(S \subset \{1, \ldots, n\}\) of size \(k\) that has high probability hits some shortest path for each pair of nodes with a short path on many edges.

Example: Node-weighted APSP

- **We divide the shortest paths into two categories**: those having lengths larger than \(n\) nodes, for some parameter \(s\), and those having lengths less than \(n\) nodes.
- **Use standard hitting set algorithm**: Suppose we want to compute the distances \(d_{uv}(s)\) for all vertices \(u, v\) that have a shortest path \(P_{uv}\) on at least \(s\) nodes. Then if we sample \(O(\log n)\) nodes \(S\) independently at random we have that with high probability for all such pairs \(u, v\), there is a node of \(P_{uv}\) in \(S\), and hence for all such \(u, v\),
  \[
  d_{uv}(s) = \min\{d_{uv}(s) + d_{s,v}\}.
  \]

Diagram of Our Quantum APSP Algorithms

(1) Compute a matrix product over some algebraic structure and iterate this matrix product \(t\) times to compute the distances between vertices of pairs that have shortest paths on few \(\leq k\) nodes. Replace brute force portion of the classical matrix product algorithms with Grover search.

(2) Pick a random sample \(S\) of size \(O(n^2/k)\) that with high probability hits some shortest path for each pair of nodes with a shortest path on many \(\geq k\) nodes. Compute the distances \(d_{u,v}(S)\) and \(d_{s,v}\) for all \(s \in S\) and \(v \in V\) using a variant of quantum Dijkstra’s algorithm particular to the shortest paths problem at hand.

The final answer is computed by taking the minimum out of all of the above answers.

Related Work

The closest related work is by Le Gall and Nishimura [3] who considered the complexity of some matrix products over semirings. They showed that using Grover search and fast rectangular matrix multiplication one can multiply two matrices over the tropical semiring in quantum \(O(n^{2+\epsilon})\) time. Unlike the results of [2], our results apply to all pairs path problems whose matrix products are not over semirings.

Another related line of work is on quantum output sensitive matrix multiplication, given two \(n \times n\) matrices whose product has at most \(L\) nonzeroes, compute their product. Le Gall [2] obtained the current best bound \(O(n^{2.473})\). Although the author never mentions this, this result also implies that the transitive closure of any graph can be computed in \(O(\min\{\sqrt{n}, L\log n\})\) quantum time, where \(L \leq n\) is the number of edges in the transitive closure.

Apart from the above mentioned consequences of prior work, our work presents the first study of all pairs path problems in the quantum setting and we are first to exhibit nontrivial, i.e. \(O(n^{2+\epsilon})\) time quantum algorithms for all pairs path problems.

References