Quantum-Accelerated Distributed Algorithms for Approximate Steiner Trees and Directed Minimum Spanning Trees

Phillip Kerger[∗,†,‡], David Bernal Neira[†,‡], Eleanor Rieffel[†]

∗ Department of Applied Mathematics, Johns Hopkins University, † Quantum Artificial Intelligence Laboratory, NASA, ‡ Research Institute of Advanced Computer Science, USRA

We present two algorithms in the Quantum CONGEST-CLIQUE model of distributed computation that succeed with high probability; One for producing an approximately optimal **Steiner Tree**, and one for producing an exact **Minimum Directed Spanning tree**. These use $\tilde{O}(n^{1/4})$ rounds of communication and $\tilde{O}(n^{9/4})$ messages, leading to a **quantum speedup in round and message complexity** compared to any known algorithms in the classical CONGEST-CLIQUE model (vs $\tilde{O}(n^{1/3})$ and $\tilde{O}(n^{7/3})$). At a high level, we achieve these results by combining classical algorithms with fast quantum subroutines. Further, these problems can not be sped up in the CONGEST (non-clique) setting, and we characterize the constants involved.

Quantum CONGEST-CLIQUE

Definition 1. The Quantum CONGEST-CLIQUE Model (qCCM) is a distributed computation model with input graph $G = (V, E, W)$ distributed over a network of n processors represented by V. Each node is assigned a unique ID number in $[n]$. Time passes in *rounds*, in which each node can:

1. Execute unlimited local computation.
2. Send a distinct message of $O(\log n)$ qubits to every other node in the network.
3. Receive all messages it is the destination for.

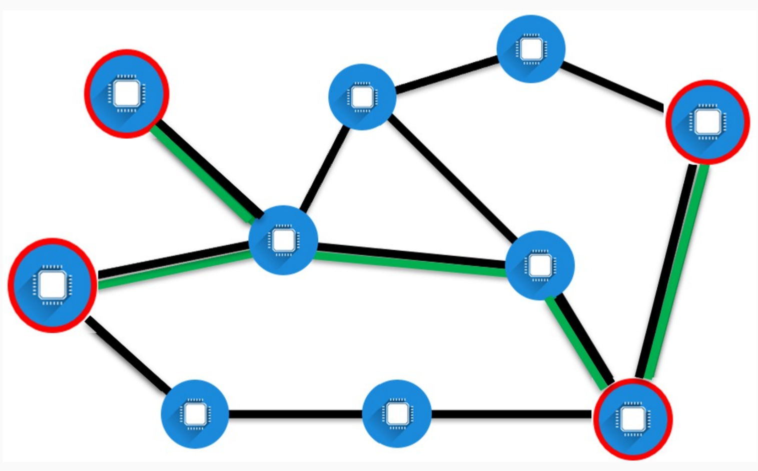
The input graph G is distributed aas follows: Each node knows its own ID number, the IDs of its neighbors in G, and the weights corresponding to the edges it is incident upon. The output solution to a problem must be given by each node $v \in V$ returning the parts of the output in its neighborhood.

Main Contributions

Theorem 1. There exists an algorithm in the qCCM that, for integer-weighted graph G, outputs a $2(1-1/l)$ approximate Steiner Tree w.p. at least $1-\frac{1}{poly(n)}$ in $\tilde{O}(n^{1/4})$ rounds of computation, where l is the number of terminal leaf nodes in the optimal Steiner Tree.

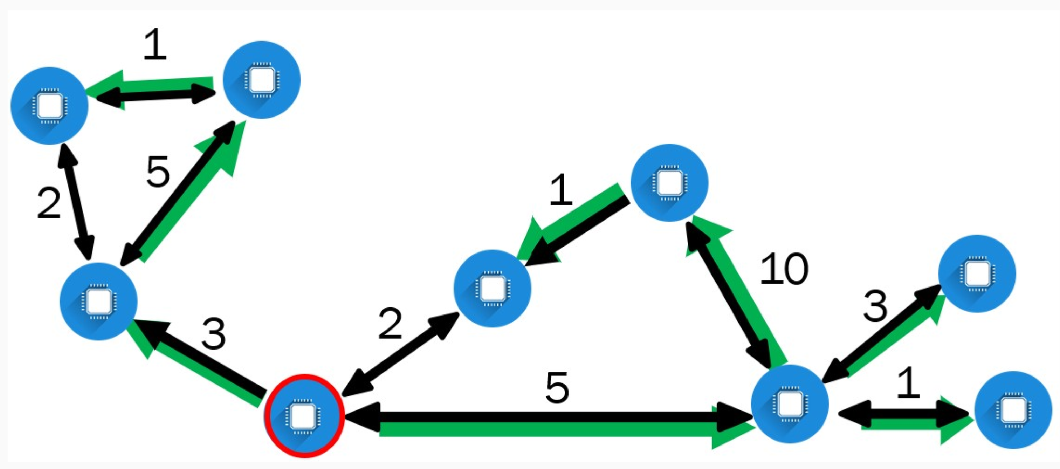
Theorem 2. There exists an algorithm in the qCCM that, for a directed and integer-weighted graph G, produces an exact Directed Minimum Spanning Tree w.p. at least $1-\frac{1}{poly(n)}$ in $\tilde{O}(n^{1/4})$ rounds of computation.

Steiner Tree



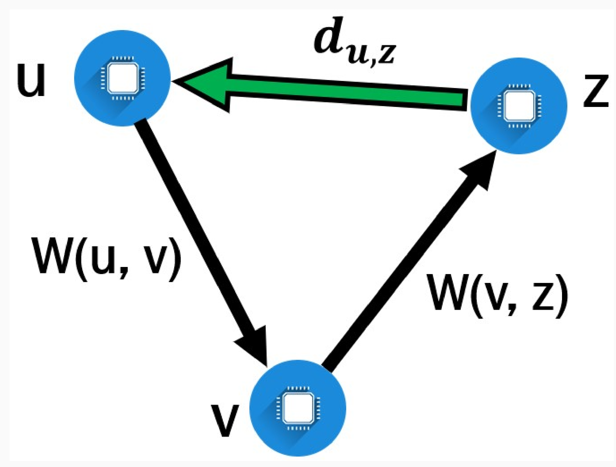
Definition 2 (Steiner Tree Problem)**.** Given a weighted, undirected graph $G = (V, E, W)$, and a set $\mathcal{Z}$ of *Steiner Terminals*, output the minimum weight tree in G that contains $\mathcal{Z}$.

Directed MST



Definition 3 (Directed Minimum Spanning Tree Problem)**.** Given a directed, weighted graph $G = (V, E, W)$ and a root node $r \in V$, output the minimum weight directed spanning tree for G rooted at r.

Distances via Negative Triangles



2-hop distances and negative triangles: The distance from u to z is less than $-d_{u,z}$ iff (u, v, z) is a negative triangle in the above graph, with added edge (z, u). Binary search on $d_{u,z}$ via finding negative triangles.

Approx Steiner Tree Algorithm

Step 1 - APSP and Routing Tables: Solve the APSP problem with an efficient routing table scheme via triangle finding in $\tilde{O}(n^{1/4})$ rounds, with success probability $(1 - 1/poly(n))$.

Step 2 - Shortest-path Forest: Construct a shortest-path forest (SPF), where each tree consists of each node's shortest path to its closest terminal. One round.

Step 3 - Weight Modifications: Modify the edge weights depending on whether they belong to a tree (set to 0), connect nodes in the same tree (set to ∞), or connect nodes from different trees (set to the shortest path distance between root terminals of the trees that use the edge). One round.

Step 4 - Minimum Spanning Tree: Construct a MST on the modified graph in $O(1)$ rounds.

Step 5 - Pruning: Prune leaves of the MST that are non-terminal nodes since these are not needed for the Steiner Tree. One round.

Distance Products and Shortest Paths

Definition 4. The *distance product* between two $n \times n$ matrices A and B is:

$$(A \star B)_{ij} = \min_k \{A_{ik} + B_{kj}\}. \tag{0.1}$$

For a $n \times n$ matrix W and an integer k, write

$$W^{k,\star} := \underbrace{W \star (W \star (...(W \star W))...)}_{k \text{ distance products}}. \tag{0.2}$$

For $G = (V, E, W)$, $W^{k,\star}_{uv}$ is the length of the shortest path from v to u using at most k hops.
Take $\lceil \log n \rceil$ distance product squares to compute n-hop (shortest path!) distances:

$$W^{2,\star} = W \star W, \quad W^4 = W^{2,\star} \star W^{2,\star}, \quad \dots \quad , W^{n,\star} = W^{2^{\lceil \log n \rceil - 1},\star} \star W^{2^{\lceil \log n \rceil - 1},\star}. \tag{0.3}$$

2-hop distances via negative triangles $\xrightarrow{O(\log n)}$ distance products $\xrightarrow{O(\log n)}$ APSP distances

Quantum Distributed Searches

Theorem 3. Let $g : X \to \{0, 1\}$, if a node u can compute $g(x)$ in r rounds in the CONGEST-CLIQUE model for any $x \in X$, then there exists an algorithm in the Quantum CONGEST-CLIQUE that has u output some $x \in X$ with $g(x) = 1$ with high probability using $\tilde{O}(r\sqrt{|X|})$ rounds.

Use this for each node to find which of its edges is part of a negative triangle for the distance products calculation!

Main References

Elkin, M., Klauck, H., Nanongkai, D., & Pandurangan, G. (2012). *Can quantum communication speed up distributed computation?* arXiv. doi: 10.48550/ARXIV.1207.5211

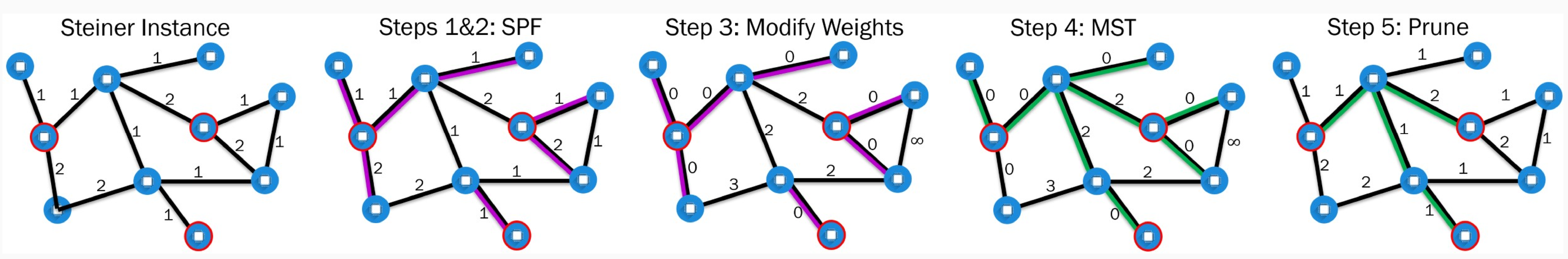
Fischer, O., & Oshman, R. (2021). A distributed algorithm for directed minimum-weight spanning tree. Distributed Computing, Springer. doi: 10.1007/s00446-021-00398-3

Izumi, T., & Gall, F. L. (2019, jul). Quantum distributed algorithm for the all-pairs shortest path problem in the CONGEST-CLIQUE model. In *Proceedings of the 2019 ACM symposium on principles of distributed computing*. ACM. doi: 10.1145/3293611.3331628

Izumi, T., Le Gall, F., & Magniez, F. (2020). *Quantum distributed algorithm for triangle finding in the congest model.* Schloss Dagstuhl - Leibniz-Zentrum für Informatik. doi: 10.4230/LIPICS.STACS.2020.23

Saikia, P., & Karmakar, S. (2019). *Distributed approximation algorithms for steiner tree in the congested clique.* arXiv. doi: 10.48550/ARXIV.1907.12011

Steiner Algorithm Visualization



DMST Algorithm

- Similarly take advantage of fast quantum shortest path calculations
- $\log n$ many single-source shortest path calculations in a vertex-contraction algorithm to get a DMST

We acknowledge funding of the USRA NASA Academic Mission Services (contract NNA16BD14C).