Davidson Institute

Fellows Scholarship

An Adaptive Hybrid Quantum Algorithm for the Metric Traveling Salesman Problem

By Arul Rhik Mazumder and Dr. Fei Li

Problem Statement and Example

Metric Traveling Salesman Problem (Metric TSP)

- Given a **complete graph** with positive **edge weights** that satisfies the **properties of a metric space**, find a cycle starting from vertex ***v*** that visits each vertex once with the **minimal edge weight sum**.

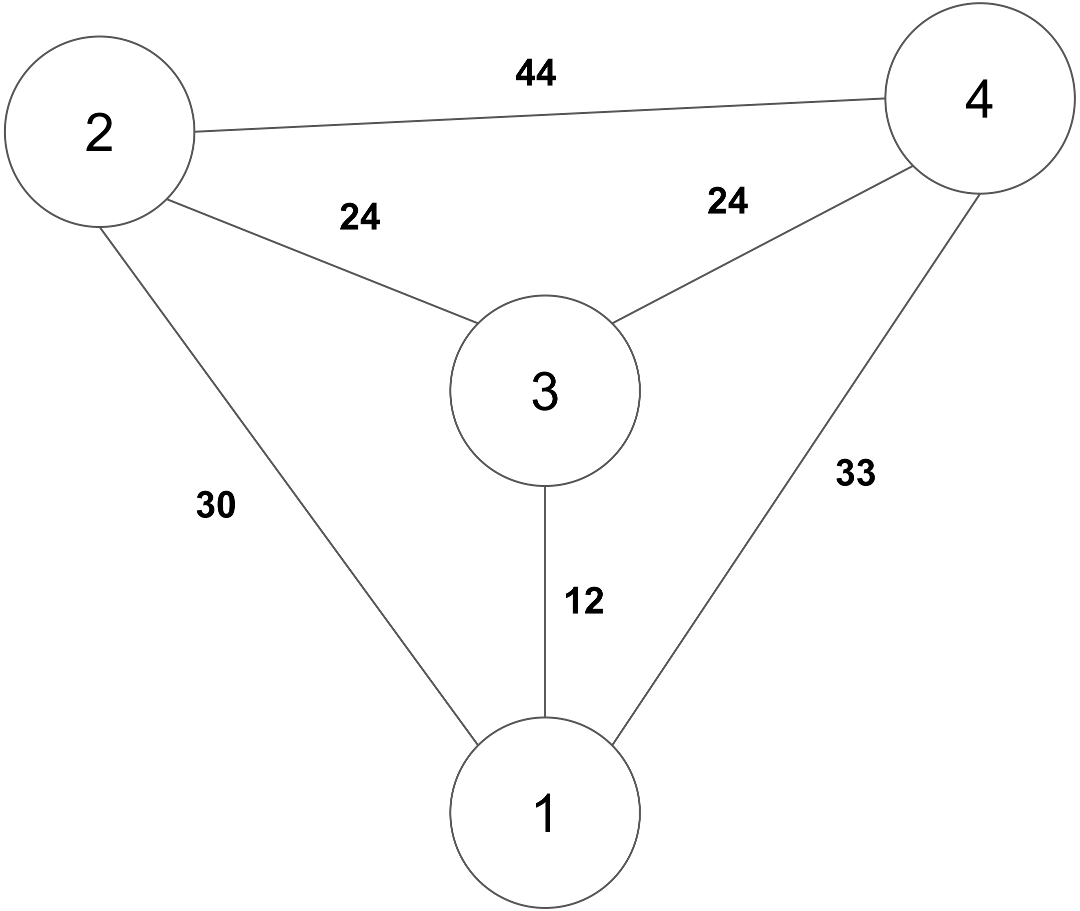
- **complete graph:** a simple undirected graph in which every pair of distinct vertices is connected by a unique edge

- **edge weight:** the value associated with an edge in the graph

- **properties of a metric space:** all distances are greater than or equal to 0, are commutative, and satisfy the triangle inequality

- **edge weight sum:** the sum of the edge weights in a path

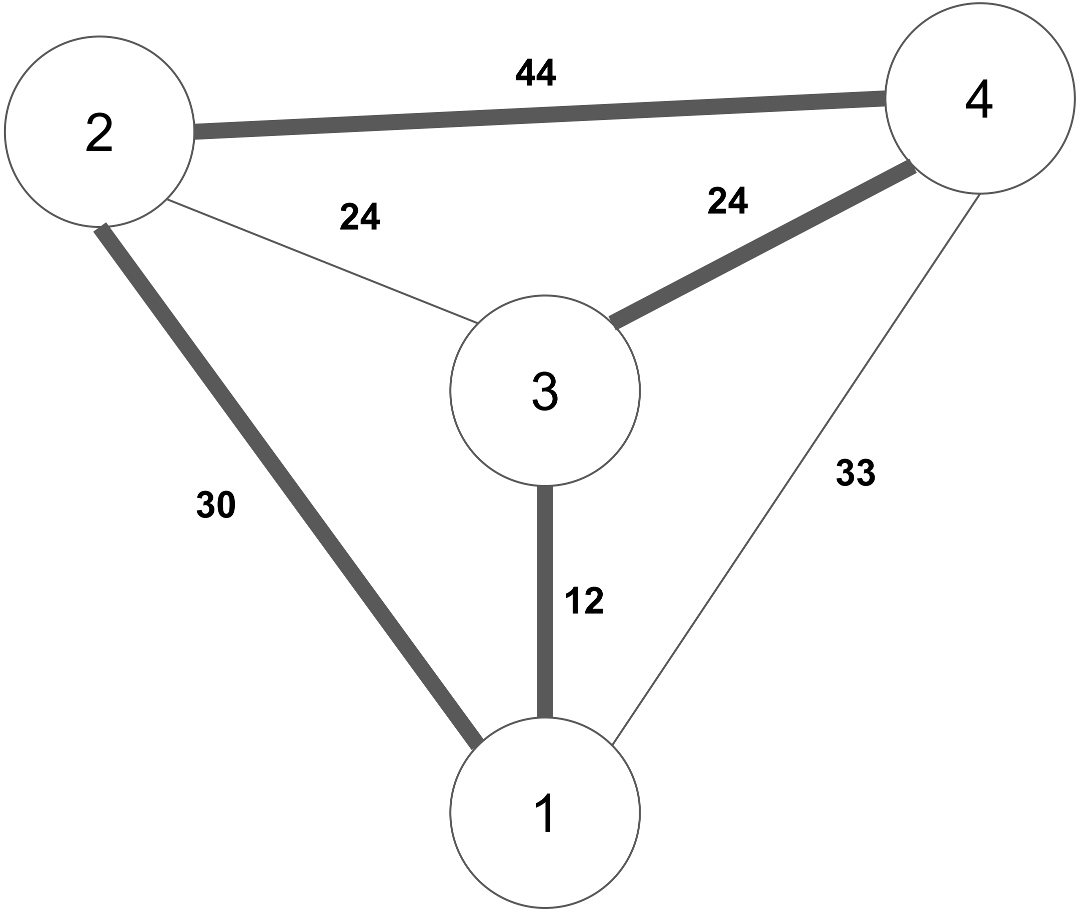
Example



On this **complete** and **metric graph.** Find a cycle with the **minimal edge weight sum.**

- Example of Metric TSP with four nodes
- **Complete** since every pair of nodes has an edge between them
- **Metric** since all distances are nonnegative and follow triangle inequality.

Example Solution



Brute-force Algorithm has time complexity of $O(n!)$

- standard Intel i7 process needs over 19 hours for a graph with only 17 nodes

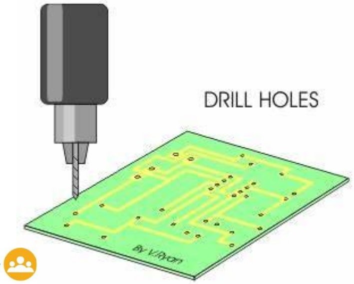
Best **Dynamic Programming** Algorithm has time complexity of $O(n^2 2^n)$

- standard Intel i7 process needs over 19 hours for a graph with only 38 nodes

Applications

Applications of Traveling Salesman Problem (TSP)

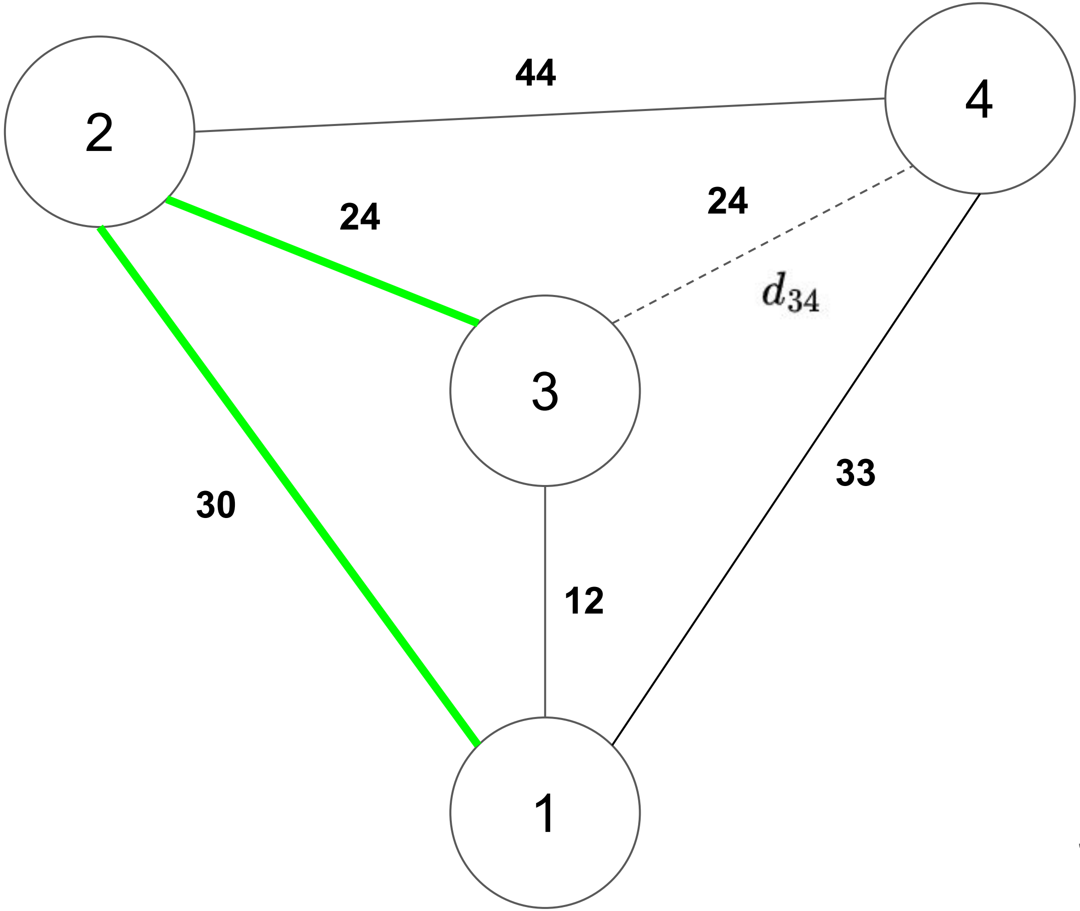
- TSP has countless applications across manufacturing, genomics and logistics.
  - orienting machine head when drilling holes in printed circuits
  - collecting goods from warehouses to transport to customers
  - positioning machinery in X-ray crystallography
  - connecting components in a computer board
  - overhauling gas turbine engines



- It has numerous other applications in various other sectors, and optimized algorithms could save countless hours and trillions of dollars

Best Classical Algorithm

Dynamic Programming Solution



Breaks the problem into $n2^n$ subproblems.

Link subproblems using the recursion $C(\{1\},1) = 0$ and

$$C(S,j) = \min_{i \in S : i \neq j} C(S - \{j\}, i) + d_{ij}$$

$C(S,j)$ represents the **minimal edge weight sum path** starting from **1** and ending at ***j*** of the subset ***S*** of nodes

d_{ij} is the edge weight connecting *i* to *j*

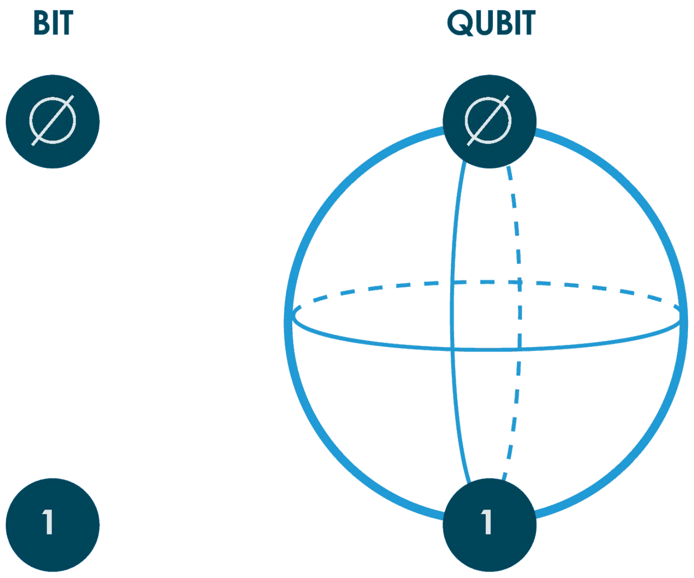
Ex.) $C(S - \{4\}, 3) + d_{34}$ on left

Stores sum in a memoization table.

Quantum Computing and the Power of Parallelism

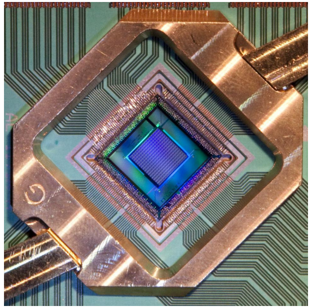
Quantum Computing

- Quantum Computing is a framework that exploits properties of quantum mechanics (superposition, entanglement, etc.) to speed-up computations
- Based on **quantum bits** or **qubits,** units that could take 0, 1 or a superposition of the two states.
- The first quantum computers were designed developed in 1980, and the first algorithm came in 1993.
- Has the potential to solve problems unsolvable for classical computers - **quantum supremacy**

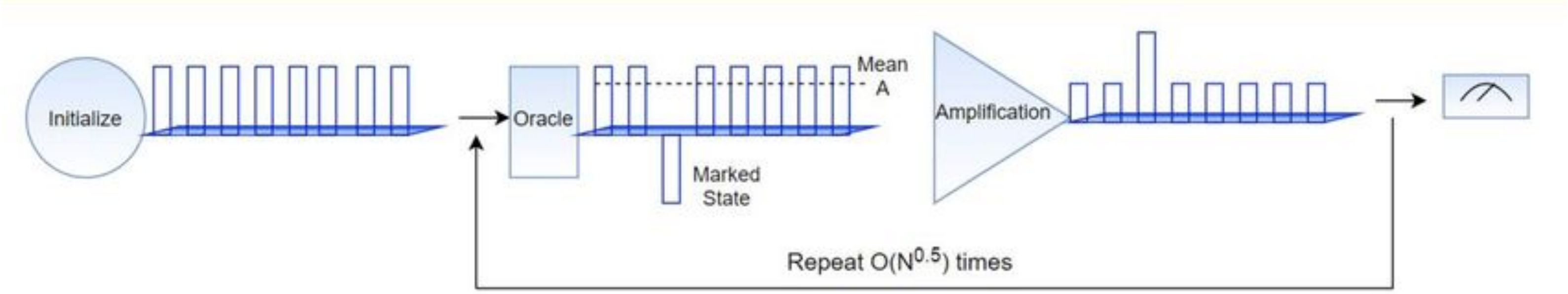


Noisy Intermediate Scale Quantum (NISQ) Era Technology

- Limited **Quantum Volume** - few high performing qubits with high error rates
- Large problems cannot be encoded and solved accurately
- For small inputs, computers can evaluate multiple inputs at once - **quantum parallelism**



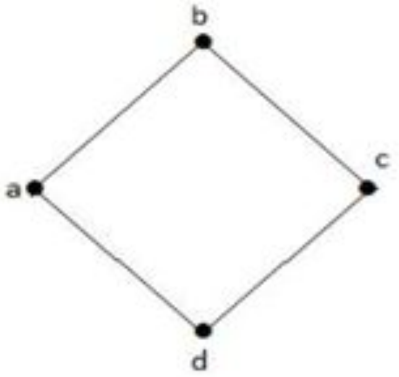
- **Grover's Algorithm:** one best current quantum algorithms, able to find a marked element in an array with size ***n*** in $O(\sqrt{n})$



Our Novel Work

Trimming the Graph

- To encode the problem on current quantum devices, we remove vertices not on the optimal subpaths
- Using the metric assumption, we mathematically prove the algorithm retains optimality
- This preprocessing step serves multiple purposes
  - ensures algorithm is adaptive, optimizing runtime based on input graph
  - decrease the classical portion of the algorithm geometrically, by reducing search space
  - reduces the input so it can be run on a Noisy Intermediate Scale Quantum (NISQ) device



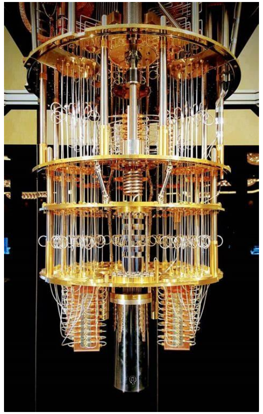
Our Adaptive Hybrid Algorithm

1. Apply **Johnson's Algorithm** to find shortest paths for subproblems

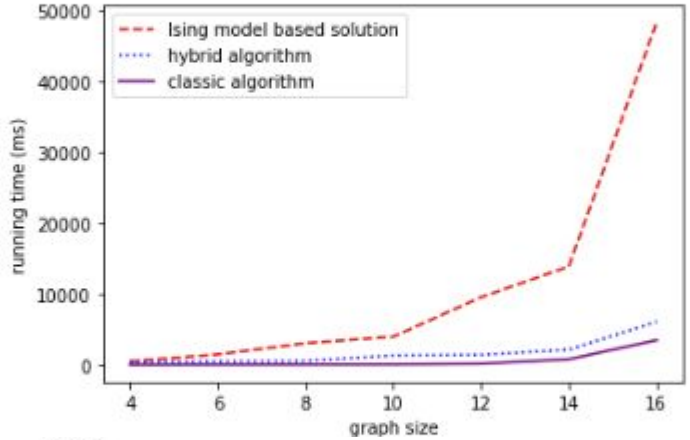
2. **Trim** the input graph to run on quantum computers

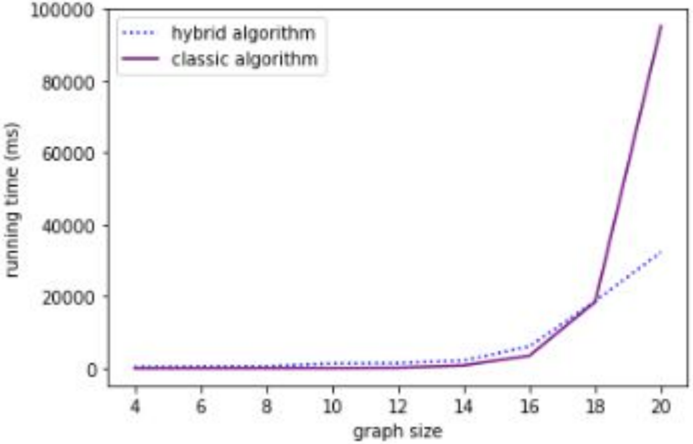
3. Search for the smallest path

- if encoded with less than 7 qubits use IBM **circuit-based** algorithm
- else use DWAVE **ising-model** based algorithm



Results and Conclusion





- Our novel hybrid work nearly matched the best classical algorithm for small inputs
- The hybrid algorithm outperformed the classical work for larger inputs and had a smaller rate of growth
  - hybrid grew by a factor of 5 while classical grew by factor of 24
  - had a lower time complexity
- Algorithmic performance will improve, as technology develops
- Approach can be applied to other NP-Hard Problems

Thanks for your Attention!
Questions?