Utility Scale Computing on NISQ

Design Problem 1

~~Logic synthesis is a key step in VLSI design, where a Boolean function is transformed into an optimized logic circuit using basic gates (AND, OR, NOT, XOR, etc.). The goal is to minimize the number of gates, power consumption, and delay while preserving the circuit's functionality. Traditional logic synthesis methods use Boolean algebra simplifications (like Karnaugh maps, Quine-McCluskey, or heuristic algorithms such as ESPRESSO). However, as circuits grow larger, these classical methods become computationally expensive. We can potentially use Quantum Approximate Optimization Algorithm (QAOA) as it is suited for logic synthesis because Boolean functions can be mapped to Quadratic Unconstrained Binary Optimization (QUBO) problems, which can easily be translated to quantum circuits.~~

~~The steps to follow would be:~~

- ~~Boolean Logic Representation~~
  - ~~Any Boolean function can be expressed in Sum of Products (SOP) or Product of Sums (POS) form~~
- ~~QUBO Encoding for Logic Synthesis~~
  - ~~Define binary variables for each possible gate configuration.~~
  - ~~Introduce cost penalties for redundant gates and large logic depth.~~
  - ~~Formulate constraints to ensure functional correctness.~~
- ~~Implementing QAOA in Qiskit for Logic Synthesis~~
  - ~~Define the QUBO Hamiltonian~~
  - ~~Solve Using QAOA~~
- ~~Evaluate performance:~~
  - ~~Run QAOA on IBM Quantum hardware~~
  - ~~Compare with classical logic minimization tools like ESPRESSO~~
  - ~~Measure circuit depth and gate count for classical vs. quantum solutions~~

~~Start with 3 and 4 variable functions, expand it to 5 variable.~~

Implement a program to solve the Max-Cut problem for a social network graph. The graph represents users as nodes and interactions (e.g., messages, likes) as edges. Your task is to partition the users into two groups such that the number of interactions between the groups is maximized. For demonstrating results on the Max-Cut problem, you can use publicly available datasets that represent graphs. One such dataset is the Zachary's Karate Club dataset, which is a classic social network dataset often used in graph analysis and community detection tasks. This is a small network with 34 nodes and 78 edges.

Input:

A social network graph represented as an adjacency list or adjacency matrix.

Each node represents a user.

Each edge represents an interaction between two users.

The graph is undirected and unweighted (for simplicity).

Output:

Two sets of users (S and T) representing the partition.

The number of interactions (edges) between the two sets.

Write a function [classical graph]to solve the Max-Cut problem for a this problem. Then use QAOA to solve the max-cut as shown in this example. Look at partitioning some large dataset ---the larger datasets may be obtained from SNAP/Kaggle. Compare/ Analyze the results of your implementation with the optimal solution (for small graphs where brute-force is feasible).

Would like to see tables of this nature:

Algorithm	Time (10 nodes)	Time (100 nodes)	Time (100 nodes)	Success Rate
Classical				
QAOA				
Optimal				

Comment on the quality of solutions obtained using QAOA.

Design Problem 2

The Subgraph Isomorphism Problem can also be formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem and solved using Quantum Approximate Optimization Algorithm (QAOA) or Variational Quantum Eigensolver (VQE). The subgraph isomorphism problem asks whether a smaller graph $G_s=(V_s,E_s)$ can be found as a subgraph within a larger graph $G_l=(V_l,E_l)$. This implies that we need to find whether there exists a mapping $f: V_s \rightarrow V_l$ that preserves edge relationships. This is an NP-complete problem, making it a good candidate for quantum optimization techniques.

The steps for this would be:

- Formulating Subgraph Isomorphism as a QUBO using binary variables x_{ij} =1 if i ε V_s is mapped to j ε V_l

- Define the QUBO Cost Function: The optimization problem can be broken into three constraints:
    - One-to-One Mapping Constraint
    - Edge Consistency Constraint
    - Uniqueness Constraint
    - The total Hamiltonian is: $H=H_{map}+H_{edge}+H_{unique}$ where each term penalizes violations of the above constraints
- Solve with QAOA or VQE

Financial fraud detection and cybersecurity anomaly detection can be formulated as a subgraph isomorphism problem, where fraudulent behavior or anomalies are represented as known subgraph patterns that we want to find within a larger network of transactions, users, or network activity. Example Graph Representation of Fraud & Anomalies.

Application	Nodes	Edges	Subgraph Pattern Example
Credit Card Fraud	Customers, Merchants, Banks	Transactions (Amount, Time, Location)	Repeated transactions between fake accounts
Money Laundering	Bank Accounts, Shell Companies	Fund Transfers	Circular transaction loops
Cyber Threats	Users, Devices, IPs	Network connections, login attempts	Suspicious lateral movement (pivoting)
Botnet Detection	Bots, C&C Servers	Communication traffic	Star-like structures or frequent short connections

For any one of the problems listed above, use some or all of the datasets and evaluate performance of QAOA on quantum hardware versus classical implementation.

Financial Fraud Detection Datasets:

Dataset	Description	Nodes / Edges
Elliptic Dataset	Bitcoin transaction graph with fraud labels.	203K nodes, 234K edges
PaySim (Synthetic)	Simulated mobile transaction fraud.	6M transactions
IEEE-CIS Fraud Detection	Real-world credit card fraud data.	300K transactions

Cybersecurity Anomaly Detection Datasets

Dataset	Description	Nodes / Edges
DARPA Intrusion Detection Dataset	Network attack dataset for cybersecurity.	5M+ network flows
CICIDS2017	Intrusion detection dataset with botnet behavior.	3.5M network events
DNS Graph Dataset	Malicious domain detection.	1M+ domain-IP connections

Would like to see tables of this nature:

Algorithm	Time (10 nodes)	Time (50 nodes)	Time (100 nodes)	Success Rate
Classical				
QAOA				

Things to turn in:

1. Neatly typed in detailed report (<10 pages)
2. Code