MASTER TEACHER'S GUIDE

Unit Title: Variational Algorithms & QAOA (Week 6)

This module shifts from "theoretical protocols" to "practical problem solving." It introduces the **Hybrid Quantum-Classical** paradigm (Variational Algorithms), which is the primary strategy for utilizing noisy (NISQ) quantum hardware. Students will implement **QAOA** to solve combinatorial optimization problems (Max-Cut).

Field	Detail
Target Audience	**Tier 3 - Undergraduate / Developer Level**
Design Principle	**Hybrid Workflow.** Concepts require students to coordinate two distinct computational resources: a quantum circuit (Ansatz) and a classical optimizer (Gradient Descent/COBYLA) via an expectation value loop.
Learning Progression	**Hybrid Loop (Variational Principle) → Cost Hamiltonian Construction → QAOA Ansatz ($$\boldsymbol{U_P, U_M}$) → Estimator & Sampler Primitives.**
Duration	**1 Week** (approx. 4×60-90 minute sessions)
Teacher Guidance	Proficiency in optimization landscapes and Python integration (*scipy.optimize* or *qiskit-algorithms*) is essential. Emphasize that the quantum computer is *only* used to estimate Energy; the classical computer drives the learning.

---

2. Pedagogical Framework: The Optimization Engine

This unit uses **Hamiltonian Physics** to define "Cost" and **Control Theory** to define "Learning." The goal is to move students from "running a circuit once" to "training a circuit iteratively."

Focus Area	Objective (The student will be able to...)	Bloom's Level
Science/Literacy	Explain the **Variational Principle**	**Understanding**
Mathematics	Map a classical graph problem (Max-Cut) to a quantum **Ising Hamiltonian** ($H_C = \sum Z_i Z_j$). Derive the unitary operators for the Cost and Mixer layers.	**Applying, Evaluating**
Computational Logic	Implement the full QAOA workflow using Qiskit **Primitives**: Use *Estimator* for the training loop and *Sampler* for retrieving the final result.	**Applying, Creating**

---

3. Computational Logic Refinements (Week 6)

A. The Cost Hamiltonian (H_C)

Concept	Explanation	Mathematical Description
Problem Mapping	Converting a graph problem into physics. Minimizing energy ≡ Maximizing cuts.	Edge (i,j) → apply with $Z_i Z_j$
Energy Penalty	Assigning high energy (+1) to "bad" states (uncut edges) and low energy (-1) to "good" states.	$Z_i Z_j$

B. The QAOA Ansatz ($U(\beta, \gamma)$)

Concept	Explanation	Mathematical Description
Phase Separator	The Cost Layer (U_P). Applies phases based on the cost function.	$U_P(\gamma) = e^{-i\gamma H_C} = \prod_{i,j} e^{-i\gamma_1 Z_i Z_j}$ (Implemented like $R_{zz}(2\gamma_1)$ on all connected qubits)
Mixer	The Mixer Layer (U_M). Allows the state to change bitstrings (explore solution space).	$U_M(\beta) = e^{-i\beta H_M} = \prod_{i} e^{-i\beta_1 X_i}$ (Implemented like $R_x(2\beta_1)$ on all connected qubits)
Layering	Repeating the process p times increases accuracy but also noise depth.	$\lvert\psi(\gamma,\beta)\rangle = U_M(\beta_p) U_P(\gamma_p) \dots U_M(\beta_1) U_P(\gamma_1) \lvert+\rangle^{\otimes n}$

C. Qiskit Primitives (Runtime V2)

Concept	Explanation
Estimator	Calculates the expectation value (average energy). Used *during* optimization.
Sampler	Measures the state to get bitstrings. Used *after* optimization to get the answer.

---

4. Exemplary Lesson Plan: Solving Max-Cut

Module: Hybrid Optimization This lesson focuses on building the full software stack required to solve a graph problem on a quantum computer.

Coding Lab: QAOA for Max-Cut

Objective	**Students will use Qiskit to define a graph, construct the Ising Hamiltonian, build the QAOA ansatz, and run a VQE-style optimization loop to find the max-cut solution.**
Required Resources	Python Environment (Jupyter), Tier3W6_codingtask.ipynb, Tier3W6.ipynb (Lecture Notes)

Step-by-Step Instructions

Part 1: The Math (Pen & Paper - Lecture Notes)

1. **Graph to Math:** Draw a simple 4-node graph. Write down the cost function $\sum(1 - Z_i Z_j)/2$ or simply $\sum Z_i Z_j$.
2. **Ansatz Logic:** Sketch the circuit. Show how R_{zz} gates connect qubits corresponding to graph edges.

Part 2: The Code (Qiskit Implementation)

1. **Task 1 (Problem):** Use *rustworkx* or *networkx* to define the graph. Visualize it.
2. **Task 2 (Hamiltonian):** Convert the graph edges into a *SparsePauliOp* (e.g., *["ZZII", "IZZI", ...]).* Verify "Good" and "Bad" states using the *Estimator*.
3. **Task 3 (Ansatz):** Use *QAOAAnsatz* from the circuit library. Transpile it for a backend (using *generate_preset_pass_manager*).
4. **Task 4 (Optimization):**
    - Define the *cost_func* that takes parameters and returns energy.
    - Use *scipy.optimize.minimize* (COBYLA) to train the parameters.
5. **Task 5 (Result):** Use the *Sampler* with optimal parameters to get the bitstring *0101* (or symmetric equivalent).

Part 3: Assessment

- **Quiz Question 2:** What is the purpose of the Cost Hamiltonian? (Answer: To map the solution to the ground state).
- **Quiz Question 5:** What is the difference between Estimator and Sampler? (Answer: Estimator = Energy/Loop, Sampler = Bitstrings/Result).
- **Quiz Question 6:** What is a "Barren Plateau"? (Answer: Flat cost landscape where gradients vanish).

---

5. Resources for Curriculum Implementation (Week 6)

Resource Name	Type	Purpose in Curriculum
Tier3W6	**Lecture Notes (IPYNB)**	Detailed derivation of the QUBO-to-Ising mapping, ansatz structure, and the logic of the hybrid loop.
Tier3W6_codingtask	**Lab Notebook (IPYNB)**	Step-by-step coding tasks to implement QAOA using Qiskit Runtime V2 primitives.
Tier3W6_quiz	**Quiz (IPYNB)**	**Knowledge Check:** 10 multiple-choice questions covering variational principles, ansatz components, and execution details.

---

6. Conclusion and Next Steps

This **Tier 3, Week 6** module introduces the modern paradigm of quantum computing: **Variational Algorithms**. Students move beyond "one-shot" circuits to "iterative training," a skill essential for Quantum Machine Learning and Chemistry.

Key Takeaway: We don't just "run" quantum algorithms; we **train** them. The quantum computer is a specialized co-processor driven by a classical optimizer.[5]

Next Steps: Week 7 will generalize this concept to **VQE & Quantum Chemistry**, applying the same hybrid loop to find the ground state energy of physical molecules (LiH) using the Jordan-Wigner mapping.