Variational Quantum Eigensolver

Joost Bus

j.c.p.bus@student.tudelft.nl

Technische Universiteit Delft, Nederland

April 29, 2020

Overview

The general idea

- Operators that describe quantum systems grow exponentially with the system size. Finding the eigenvalues of such operators gets difficult very fast.
- Quantum Phase Estimation (QPE) can efficiently find the eigenvalue of a given eigenvector but requires fully coherent evolution.
- Variational Quantum Eigensolver (VQE) is an alternative algorithm that seeks to find (the lowest) eigenvalue of an operator, and its corresponding eigenstate using a hybrid algorithm which quantum parts are low depth and therefore requires less coherence.

Areas of application

- Chemistry *original 2013 paper was about finding the ground state of $He\text{-}H^+$*
- Optimisation *that's what we will be using it for*

Underlying mathematics

Any hamiltonian $\mathcal{H}$ can be written as

$$\mathcal{H} = \sum_{i,\alpha} h^i_\alpha \sigma^i_\alpha + \sum_{i,j,\alpha,\beta} h^{i,j}_{\alpha,\beta} \sigma^i_\alpha \sigma^j_\beta + \dots \tag{1}$$

for real h where Roman indices identify the subspace on which the operator acts, and Greek indices identify the Pauli operator.

$\Longrightarrow$ **we can break a Hamiltonian down in the sum of products of Pauli matrices**

Underlying mathematics

Therefore we can use linearity

$$\langle \mathcal{H} \rangle = \sum_{i,\alpha} h^i_\alpha \langle \sigma^i_\alpha \rangle + \sum_{i,j,\alpha,\beta} h^{i,j}_{\alpha,\beta} \langle \sigma^i_\alpha \sigma^j_\beta \rangle + \dots \tag{2}$$

Note It might happen that we need exponentially many terms to construct a certain Hamiltonian. However, many physical systems only need polynomially many terms, such as

- the electronic structure Hamiltonian of quantum chemistry
- the quantum Ising Model
- the Heisenberg Model

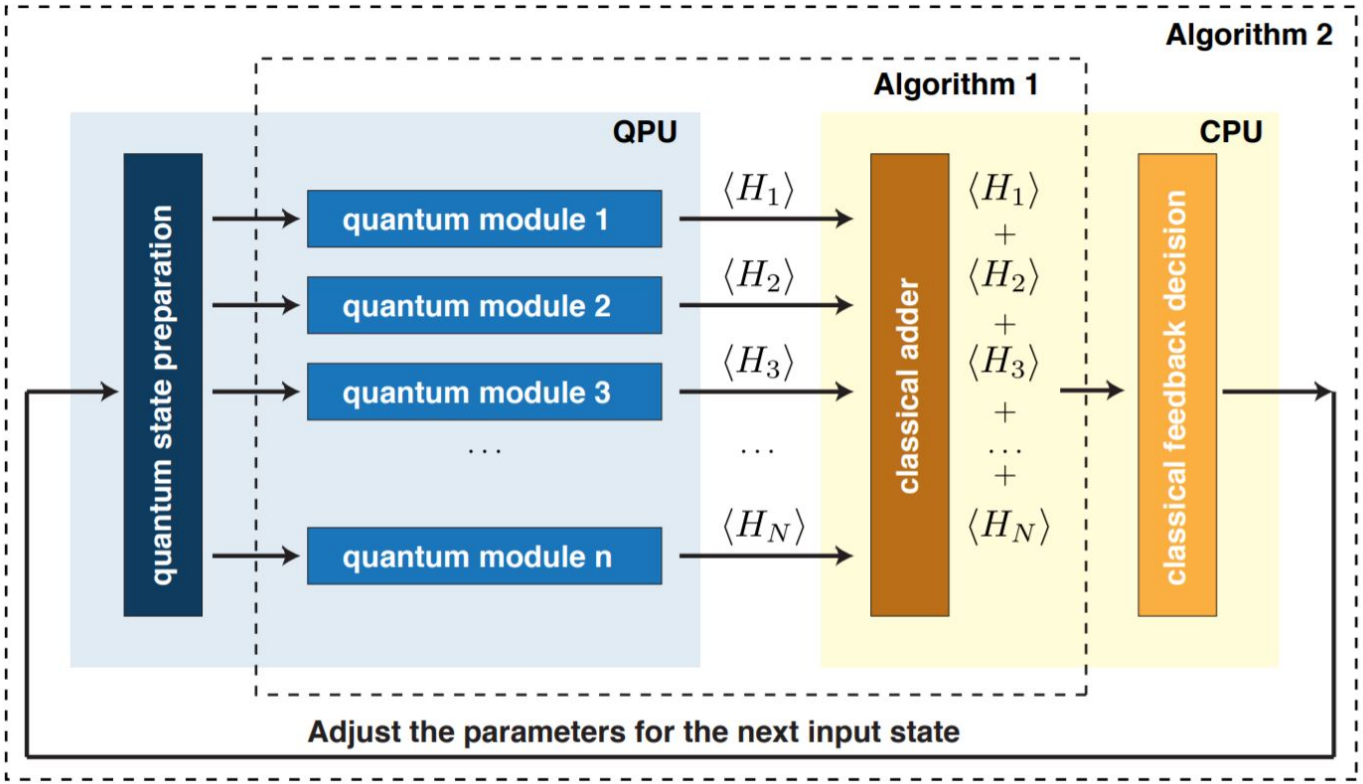

Figure: Peruzzo et al. (2013)

Rayleigh quotient

The eigenvalue problem can be restated as a variational problem on the **Rayleigh quotient**

$$|\psi\rangle_{\text{ground}} = \arg\min \frac{\langle\psi|\mathcal{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{3}$$

Where $|\phi\rangle$ depends on some set of parameters $\{\theta_i\}$

Classical part - optimisation

Algorithm 1 is used as a quantum subroutine of Algorithm 2 (better known as VQE). Also check appendix on p. 7 of Peruzzo et al. (2013)

Often the choice is Nelder-Mead (from what I encountered), rather than gradient descent because NM is more robust to non-smooth landscapes.

VQE and QAOA

In QAOA we construct a parametrised circuit, based on a cost Hamiltonian $\mathcal{H}_C$, encoding the problem, and a mixer Hamiltonian $\mathcal{H}_B$. From these Hamiltonians we construct unitary operators

$$U(\mathcal{H}_I, \alpha) := e^{-i\alpha\mathcal{H}_I} \tag{4}$$

where α is some angle and I either designates C (cost) or B (mixer).

VQE and QAOA

Using these definitions, we can lay out the steps of QAOA

1. Start with an initial state; the equal superposition over all bitstrings

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{b \in \{0,1\}} |b\rangle \tag{5}$$

2. Evolve with the cost Hamiltonian using $U(\mathcal{H}_C, \gamma_1)$ over angle γ
3. Evolve with the mixer Hamiltonian using $U(\mathcal{H}_C, \beta_1)$ over angle β
4. Repeat the previous two steps with different angles γ_i and β_i respectively, do this p times. This results in the state

$$|\vec{\gamma}, \vec{\beta}\rangle := \Pi_{i=1}^{p} U(\mathcal{H}_B, \beta_i) U(\mathcal{H}_C, \gamma_i) \tag{6}$$

VQE and QAOA

5. We are looking for $2p$ angles $\gamma_1, \ldots \gamma_p, \beta_1, \ldots \beta_p$ in order to optimise the expectation of the cost Hamiltonian $\mathcal{H}_C$

$$F_p := \langle \vec{\gamma}, \vec{\beta} | \mathcal{H}_C | \vec{\gamma}, \vec{\beta} \rangle \tag{7}$$

Note we can determine these angles however we want. Also note that this is in fact very similar to equation 3, namely we are trying to find the ground state of a given operator

$$\Longrightarrow \text{ we can use VQE!}$$

6. From the state (6) determined by these optimzal angles we sample bitstrings. The most probable encodes the approximate optimum.

Some remarks and things to look into

- **Overkill** Currently, for the small graphs we can encode are more easily solved using brute force.
- **Topology** I am currently using a simulator / QVM. The topology of actual quantum chips is completely ignored. This is especially relevant for higher p
- **Noise** I have yet to look at the influence on noise on both VQE and QAOA. Probably there is already quite some literature on that. It is probably possible to simulate the noise as well
- **Classical comparison** It may be interesing to look at the performance compared to the best classical algorithm, namely Goemans-Williamson.
- **Why use QAOA at all when you can use VQE?** Apparently you can also VQE to solve Max-Cut problems without using QAOA, have a look here: `https://www.quanta.guru/docs/optimization/salesman/salesman/`
- **Scaling of p** How will the problem scale with p?