# QAOA for problems in linear algebra

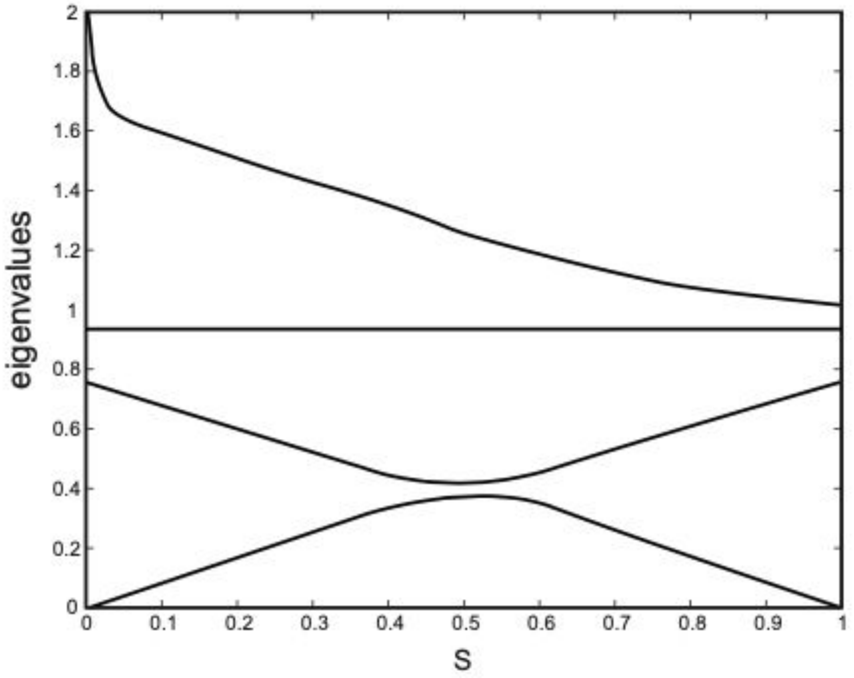
2020/03/07 - Blueqat Tokyo Summit 2020

Gaurav Singh (Blueqat)

## Background : Adiabatic Quantum Computing

$$\mathcal{H}(t) = s(t)\,\mathcal{H}_I + (1 - s(t))\,\mathcal{H}_{dis}.$$

1. An initial Hamiltonian $H_I$ chosen so that the ground state of the system is easy to find.
2. A final (also called problem) Hamiltonian $H_F$ that encodes the objective function so that the ground state is an eigenstate of $H_F$ having minimum eigenvalue. That is, a ground state of the quantum register corresponds to an optimal solution to P .
3. An adiabatic evolution path, a function s(t) that decreases from 1 to 0 as t goes from 0 to tf, for some elapsed time $t_f$ .
4. In principle if the evolution is slow enough, we are able to find a solution to any problem if we can encode its solution in the ground state of Hc.



Assume that $H_s$ has a nondegenerate ground state throughout time s. Let ρ denote the spectral gap, the (positive) difference between the eigenvalues of the ground state and of the first excited state, at any time s. Let $\rho_m$ be the minimum spectral gap, then the time for the whole process

$$\tau(s) \gg \frac{\left\|\frac{d}{ds}\widetilde{\mathcal{H}}(s)\right\|}{(\delta_m)^2},$$

## Quadratic Unconstrained Binary Optimization (QUBO)

The QUBO Objective function is as follows :

$$F(q) = \sum_{a} v_a q_a + \sum_{a<b} w_{ab} q_a q_b$$

## Ising Objective Function

One popular method of encoding an optimization problem to be solved using QAOA, is to first formulate the problem as an Ising Objective function :

$$F(\sigma) = \sum_{a} h_a \sigma_a + \sum_{a<b} J_{ab} \sigma_a \sigma_b$$

$$\text{where } \sigma_a = 2q_a - 1$$

# Quantum Approximate Optimization Algorithm

For us to to conveniently work using QAOA we need to map the Ising objective function to the qubits for which we use the Quantum Ising Hamiltonian which serves as the objective function of the QAOA algorithm.

$$\hat{C} = \sum_{a} h_a \hat{\sigma}_a^{(z)} + \sum_{a<b} J_{ab} \hat{\sigma}_a^{(z)} \hat{\sigma}_b^{(z)}$$

The other type of Hamiltonian in the QAOA process is a summation of individual Pauli X operators for each qubit involved in the process, which intuitively represents a transverse field in the Ising model:

$$\hat{B} = \sum_{a} \hat{\sigma}_a^{(x)}$$

In QAOA, the qubits are initially put in a uniform superposition over the computational basis states using the Hadamard gate.

$$\frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle .$$

Next we apply the Hamiltonian pair C and B, p number of times which are governed by the parameters **λ and β** where $\gamma_l \in [0,2\pi]$ and $\beta_l \in [0,\pi]$. Taking the initial state as a uniform superposition state, the state after p steps come out to be:

$$|\gamma,\beta> = U(B,\beta)U(C,\gamma)p\ldots\ldots U(B,\beta)U(C,\gamma)_2U(B,\beta)U(C,\gamma)_1|\psi> ,$$

where the unitary operators U(B,β) and U(C,γ) evolve the state with the Hamiltonian H_B and H_C over time β and γ.

The expectation value of the objective Hamiltonian with respect to the state as **|γ,β>** is given as :

$$\langle \gamma,\beta | C | \gamma,\beta \rangle$$

**We use classical optimization algorithms to optimize the parameters β and γ of length p taking the expectation value as the cost function for the minimization.**

**As long as the depth of the circuit p << n where n is the number of the qubits involved , we are efficiently able to sample the best solution of the ground state.**

## QAOA for solving linear equations with binary solutions

Let’s say we have a linear equation : Ax = b and we want to solve for x, where $A \in R^{m \times n}$ , $x \in \{0,1\}^n$ and $b \in R^m$ with $m > n$.

We use the binary linear least squares problem to get the optimum value of x.

$$\arg\min_x ||Ax - b||$$

The motivation behind choosing the BLLS problem to be applied to QAOA is twofold: firstly, its an NP-Hard problem that makes it a suitable candidate for QAOA. Secondly,it can act as a building block for other hard problems in linear algebra, such as the Non-negative Binary Matrix Factorization

$$Ax - b = \begin{pmatrix} A_{11} & A_{12} & ... & A_{1n} \\ A_{21} & A_{22} & ... & A_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1} & A_{m2} & ... & A_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$Ax - b = \begin{pmatrix} A_{11}x_1 + A_{12}x_2 + ... + A_{1n}x_n - b_1 \\ A_{21}x_1 + A_{22}x_2 + ... + A_{2n}x_n - b_2 \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + ... + A_{mn}x_n - b_m \end{pmatrix}$$

$$\|Ax - b\|_2^2 = \sum_{i=1}^{m} (A_{i1}x_1 + A_{i2}x_2 + ... + A_{in}x_n - b_i)^2$$

QUBO formulation for the Binary Linear Least Squares problem :

$$F(q) = \sum_{j} v_j q_j + \sum_{j<k} w_{jk} q_j q_k$$

For us to work with QAOA we need to convert it into Ising Formulation :

$$F(\sigma) = \sum_{j} h_j \sigma_j + \sum_{j<k} J_{jk} \sigma_j \sigma_k$$

The objective Hamiltonian in the form of ising Hamiltonian function comes out to :

$$\hat{C} = \sum_{j} h_j \hat{\sigma}_j^{(z)} + \sum_{j<k} J_{jk} \hat{\sigma}_j^{(z)} \hat{\sigma}_k^{(z)}$$

We create Unitary operators which act as gate for the gate model out of both the objective Hamiltonian **C** and the mixer hamiltonian **B** which is only possible because of the commutation of the individual components in both the Hamiltonians.

$$e^{-i\gamma_l \hat{C}} = \prod_j e^{(-ih_j\gamma_l)\hat{\sigma}_j^{(z)}} \prod_{j<k} e^{(-iJ_{jk}\gamma_l)\hat{\sigma}_j^{(z)}\hat{\sigma}_k^{(z)}}$$

$$e^{-i\beta_l \hat{B}} = \prod_j e^{(-i\beta_l)\hat{\sigma}_j^{(x)}}$$

These are time evolution operators which get their origin from the Schrodinger Equation. To understand these better kindly visit the given article on Blueqat Blog.

https://medium.com/mdr-inc/time-evolution-operators-be2968493b86

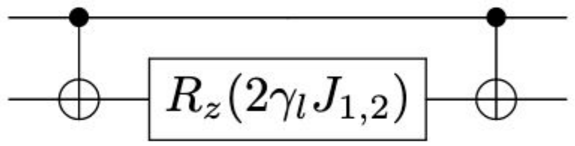
The unitary gates formed out of the objective Hamiltonian C and the mixer Hamiltonian B have **Pauli X** and **Pauli Z** components, thus we can write them as rotation operators about the Pauli gates.

$$R_x(\omega) = e^{-i\frac{\omega}{2}\hat{\sigma}^{(x)}} = \begin{pmatrix} \cos\omega/2 & -i\sin\omega/2 \\ -i\sin\omega/2 & \cos\omega/2 \end{pmatrix}$$

$$R_z(\omega) = e^{-i\frac{\omega}{2}\hat{\sigma}^{(z)}} = \begin{pmatrix} e^{-i\omega/2} & 0 \\ 0 & e^{i\omega/2} \end{pmatrix}$$

The two qubit interaction ZZ gate can be implemented in the following form.

$$e^{(-iJ_{1,2}\gamma_l)\hat{\sigma}_1^{(z)}\hat{\sigma}_2^{(z)}} =$$



Thus the ansatz for the QAOA problem comes out to be U_C and U_B which helps us span across the entire Bloch Sphere.

Finally we find the expectation value of the objective Ising Hamiltonian of the Binary Linear Least Square about the newly created states using the above ansatz which is done on the quantum computer.

On the classical computer we optimize the angles $\lambda$ and $\beta$ using any of the classical optimization methods which are then again input in the above ansatz to get the minimum expectation value or the ground state of the Objective Hamiltonian.

As an example, let's take the linear equation : Ax = b ,

Where A =

| | | |
|---|---|---|
| 1 | 2 | 1 |
| -1 | 1 | -1 |
| 1 | 1 | 1 |

x =

| |
|---|
| q1 |
| q2 |
| q3 |

And the solution vector b is :

| |
|---|
| 3 |
| 0 |
| 2 |

We need to find the binary solution of x which would satisfy the above linear equation.

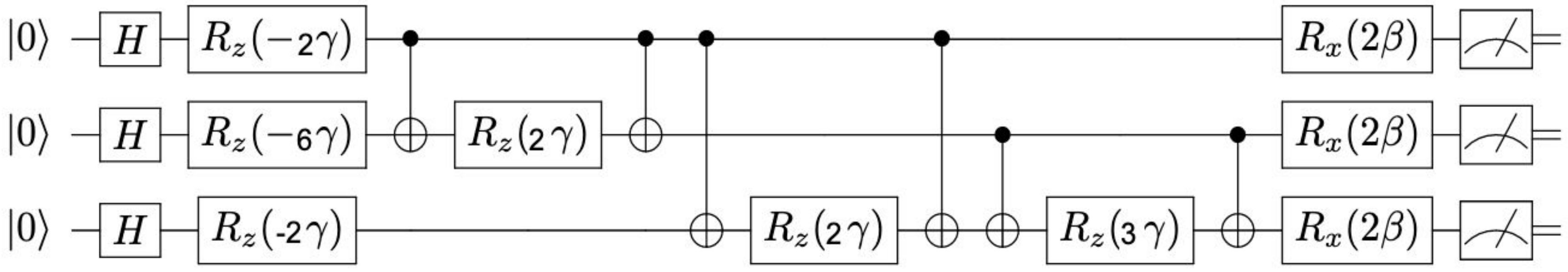
The QUBO formulation of the BLLS problem which will help in solving the linear equation comes out to be :

$$-7q_1 - 10q_2 - 7q_3 + 4q_1q_2 + 6q_1q_3 + 4q_2q_3$$

THe Ising Hamiltonian formulation comes out to be :

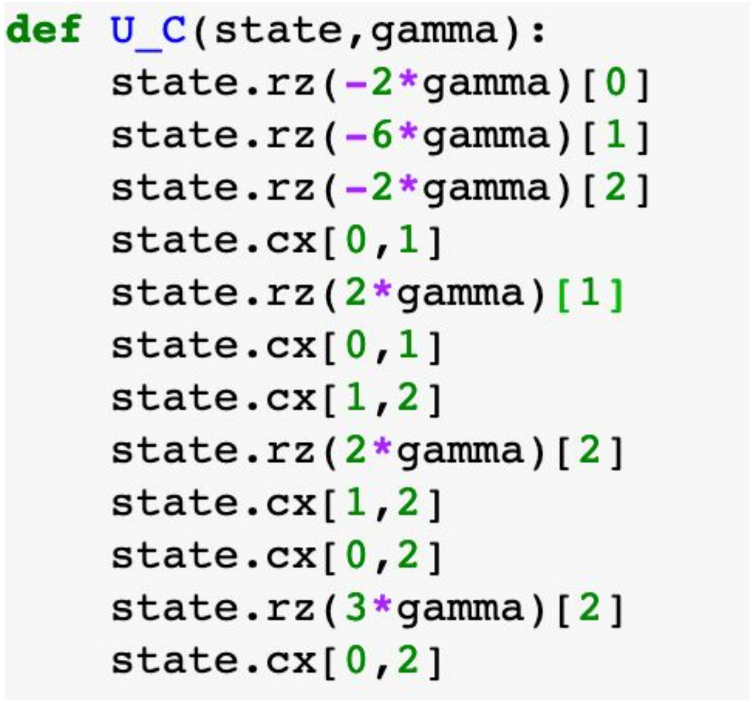
$$-\sigma_1 - 3\sigma_2 - \sigma_3 + \sigma_1\sigma_2 + \sigma_2\sigma_3 + 1.5\sigma_1\sigma_3,$$ **where** $\sigma_i = \sigma_{i(z)}$

The Circuit for the above problem is :

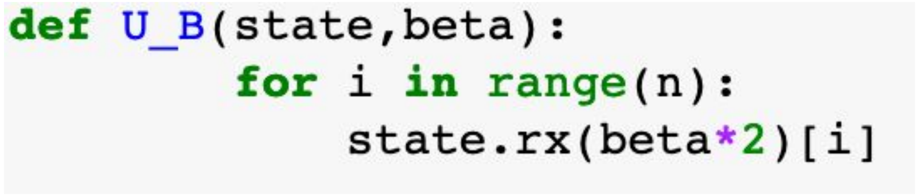


The Blueqat code for the state preparation of the above circuit is :

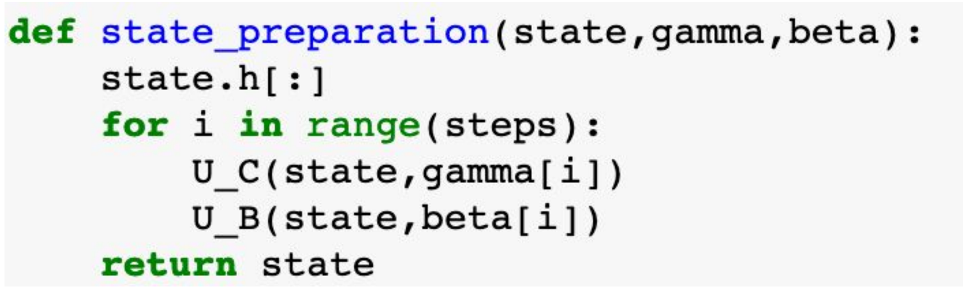
from blueqat import Circuit



```
def U_C(state,gamma):
    state.rz(-2*gamma)[0]
    state.rz(-6*gamma)[1]
    state.rz(-2*gamma)[2]
    state.cx[0,1]
    state.rz(2*gamma)[1]
    state.cx[0,1]
    state.cx[1,2]
    state.rz(2*gamma)[2]
    state.cx[1,2]
    state.cx[0,2]
    state.rz(3*gamma)[2]
    state.cx[0,2]
```

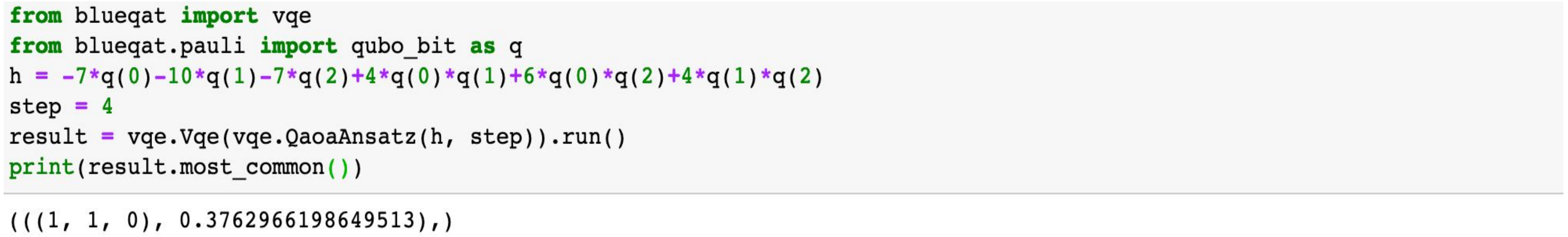


```
def U_B(state,beta):
        for i in range(n):
            state.rx(beta*2)[i]
```



```
def state_preparation(state,gamma,beta):
    state.h[:]
    for i in range(steps):
        U_C(state,gamma[i])
        U_B(state,beta[i])
    return state
```

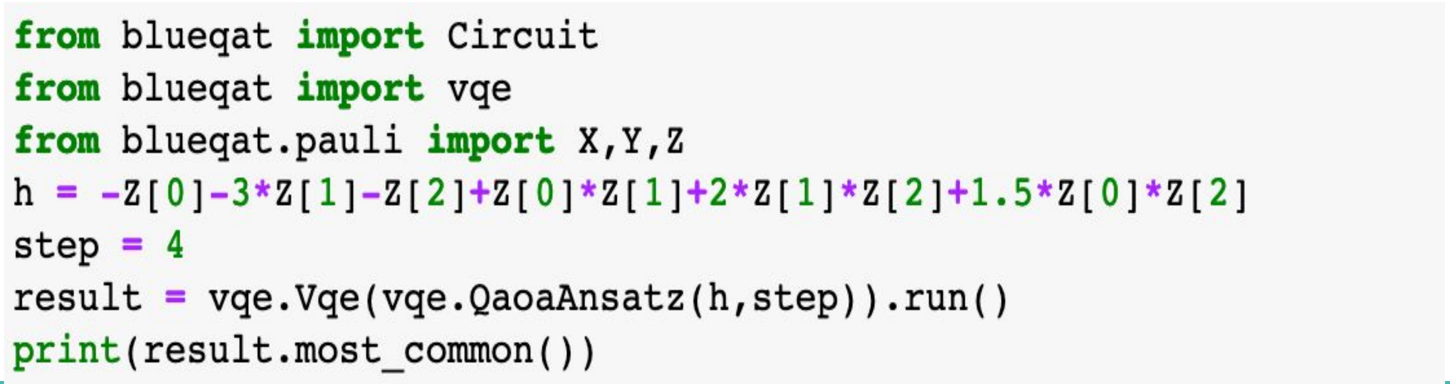
Blueqat has a function for finding the optimum solution for combinatorial problem using QAOA. It inputs both QUBO format and the Ising Hamiltonian format. The Blueqat code for the given linear algebra problem using QUBO input is :



```
from blueqat import vqe
from blueqat.pauli import qubo_bit as q
h = -7*q(0)-10*q(1)-7*q(2)+4*q(0)*q(1)+6*q(0)*q(2)+4*q(1)*q(2)
step = 4
result = vqe.Vqe(vqe.QaoaAnsatz(h, step)).run()
print(result.most_common())
```

```
(((1, 1, 0), 0.3762966198649513),)
```

The code with the Ising Hamiltonian input is :

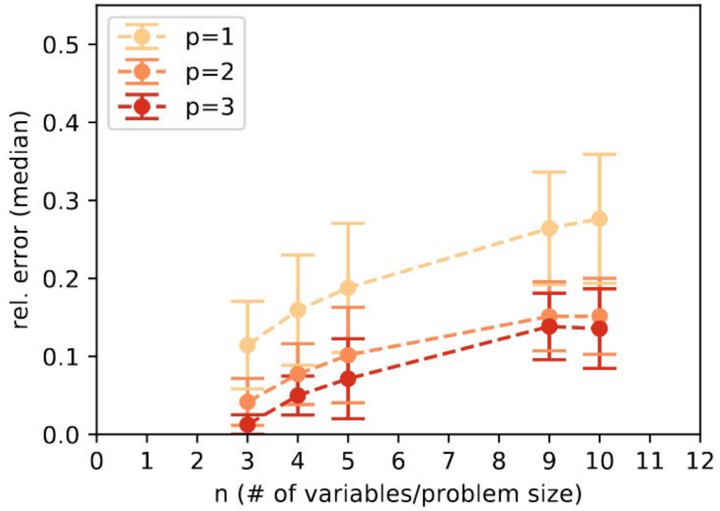


```
from blueqat import Circuit
from blueqat import vqe
from blueqat.pauli import X,Y,Z
h = -Z[0]-3*Z[1]-Z[2]+Z[0]*Z[1]+2*Z[1]*Z[2]+1.5*Z[0]*Z[2]
step = 4
result = vqe.Vqe(vqe.QaoaAnsatz(h,step)).run()
print(result.most_common())
```

## Relative Error

We can calculate the relative error of the expectation value with respect to the ground energy as :

$$\text{relative error} = \left| \frac{\text{expectation energy} - \text{ground state energy}}{\text{ground state energy}} \right|$$



https://arxiv.org/pdf/2006.15438.pdf

## Challenges

1. One theoretical challenge is the proper pre-processing of the problem Hamiltonian by scaling and shifting the coefficients of the objective function, such that we optimally make use of the parameter space $\beta \in [0,\pi]$, $\gamma \in [0,2\pi]$.
2. Computationally non-trivial problems typically require a high degree of graph connectivity . Simultaneously, a high connectivity poses a challenge in quantum chip implementation because not all gatesets implement non-nearest neighbour interactions natively.

## References :

1. Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice - Morgan & Claypool books
2. QAOA for hard problems : https://arxiv.org/pdf/2006.15438.pdf
3. Time Evolution Operators Time Evolution Operators - Blueqat Developer (MDR Inc.)
4. https://medium.com/@gerrard.gaurav/qaoa-explained-with-max-example-632a3f101f12