CSC

ICT Solutions for Brilliant Minds

Introduction to quantum computing and FiQCI - QAOA

Olli Mukkula – Application specialist at CSC

Quantum Approximate Optimization Algorithm Introduction

- What is it?
  - QAOA is a quantum general-purpose algorithm that is conceived to solve combinatorial optimization problems on <u>NISQ processors</u>

Quantum Approximate Optimization Algorithm Introduction

- What is it?
  - QAOA is a quantum general-purpose algorithm that is conceived to solve combinatorial optimization problems on NISQ processors
- Motivation
  1. Resistance to noise: QAOA can be implemented with low-depth circuits and the algorithm seems to be robust against noise beyond what would be expected for low-depth circuits.

Quantum Approximate Optimization Algorithm Introduction

- What is it?
  - QAOA is a quantum general-purpose algorithm that is conceived to solve combinatorial optimization problems on NISQ processors
- Motivation
  1. Resistance to noise: QAOA can be implemented with low-depth circuits and the algorithm seems to be robust against noise beyond what would be expected for low-depth circuits.
  2. General purpose: QAOA has potentially a wide range of applications, because it solves combinatorial optimization problems; which includes a huge amount of different problems.

Quantum Approximate Optimization Algorithm Introduction

- The idea of QAOA is to encode a classical problem into the **energy landscape** of a quantum system. The QAOA algorithm consists of the following steps:

Quantum Approximate Optimization Algorithm Introduction

- The idea of QAOA is to encode a classical problem into the **energy landscape** of a quantum system. The QAOA algorithm consists of the following steps:
    1. Define a classical cost function, which has the problem solution encoded into it.

Quantum Approximate Optimization Algorithm Introduction

- The idea of QAOA is to encode a classical problem into the **energy landscape** of a quantum system. The QAOA algorithm consists of the following steps:
    1. Define a classical cost function, which has the problem solution encoded into it.

    2. Map into quantum cost function (Hamiltonian). The expectation value of the Hamiltonian gives the energy of a quantum system.

Quantum Approximate Optimization Algorithm Introduction

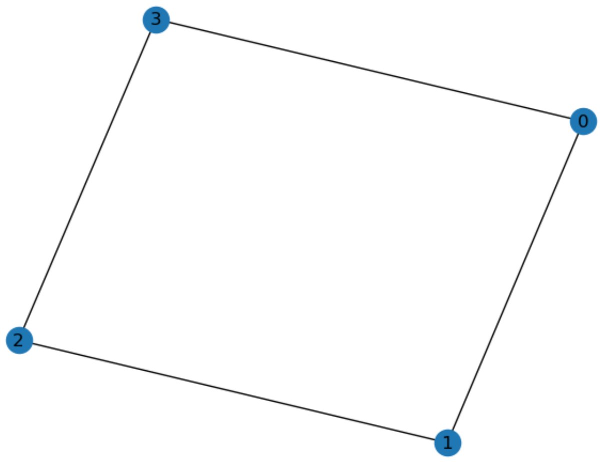
- The idea of QAOA is to encode a classical problem into the **energy landscape** of a quantum system. The QAOA algorithm consists of the following steps:
  1. Define a classical cost function, which has the problem solution encoded into it.
  2. Map into quantum cost function (Hamiltonian). The expectation value of the Hamiltonian gives the energy of a quantum system.
  3. Construct an anzatz quantum circuit, and operate it with our Hamiltonian. Calculate the expectation value of the Hamiltonian to get the energy.

Quantum Approximate Optimization Algorithm Introduction

- The idea of QAOA is to encode a classical problem into the **energy landscape** of a quantum system. The QAOA algorithm consists of the following steps:
    1. Define a classical cost function, which has the problem solution encoded into it.
    2. Map into quantum cost function (Hamiltonian). The expectation value of the Hamiltonian gives the energy of a quantum system.
    3. Construct an anzatz quantum circuit, and operate it with our Hamiltonian. Calculate the expectation value of the Hamiltonian to get the energy.
    4. Use classical optimizer to find quantum ansatz that minimizes the energy of the system.

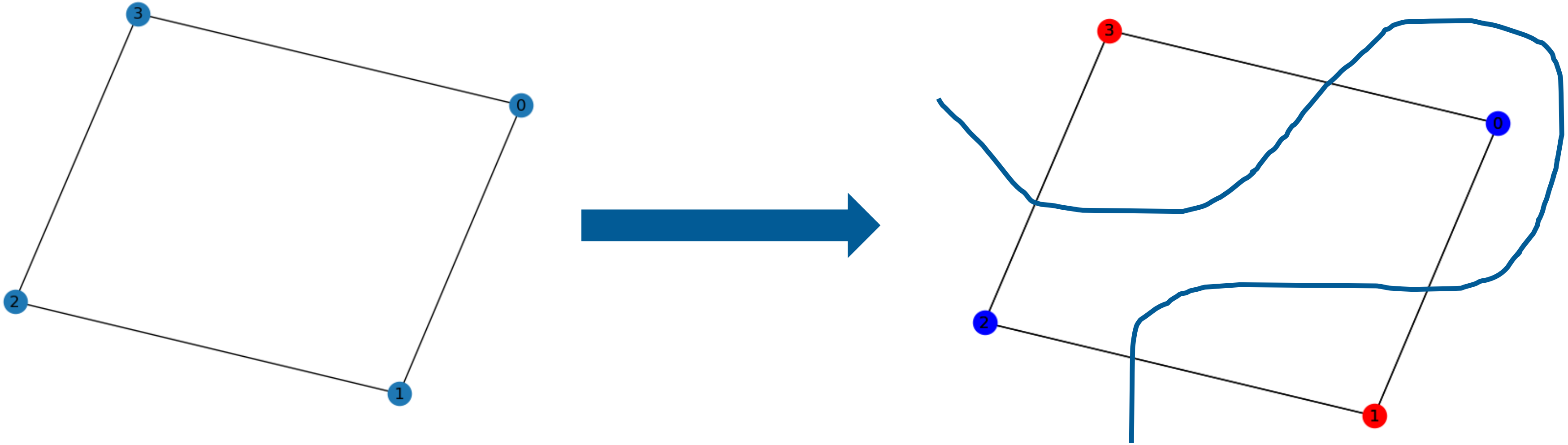
Quantum Approximate Optimization Algorithm Introduction

- Example application: The MaxCut problem
  - *Divide vertices of a graph into two complementary sets, such that the number of edges between the partitions is as high as possible*



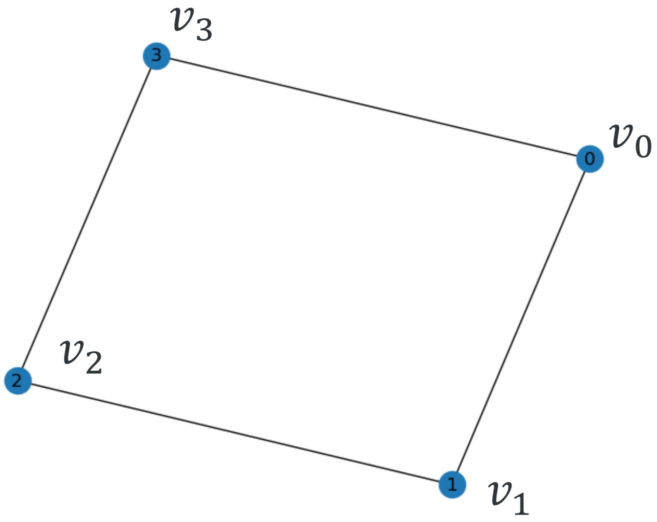
Quantum Approximate Optimization Algorithm Introduction

- Example application: The MaxCut problem
  - *Divide vertices of a graph into two complementary sets, such that the number of edges between the partitions is as high as possible*



Quantum Approximate Optimization Algorithm Introduction

- MaxCut cost function
  1. Assign variable $v_i \in \{0,1\}$ to each vertex, that defines whether the vertex belongs to 0-partition or 1-partition

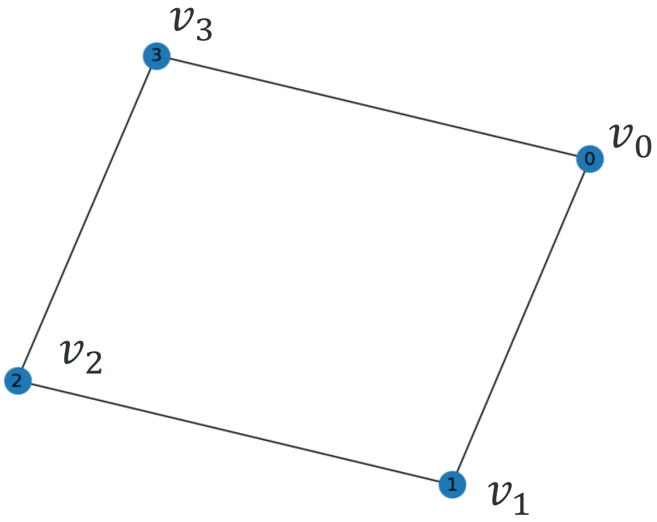


Quantum Approximate Optimization Algorithm Introduction

- MaxCut cost function
  1. Assign variable $v_i \in \{0,1\}$ to each vertex, that defines whether the vertex belongs to 0-partition or 1-partition

  2. Define a cost function, which encodes the solution to the problem

$$C = \underbrace{v_0 \wedge v_1}_{\substack{0 \text{ if } v_0 = v_1 \\ 1 \text{ if } v_0 \neq v_1}} + v_1 \wedge v_2 + v_2 \wedge v_3 + v_0 \wedge v_3$$



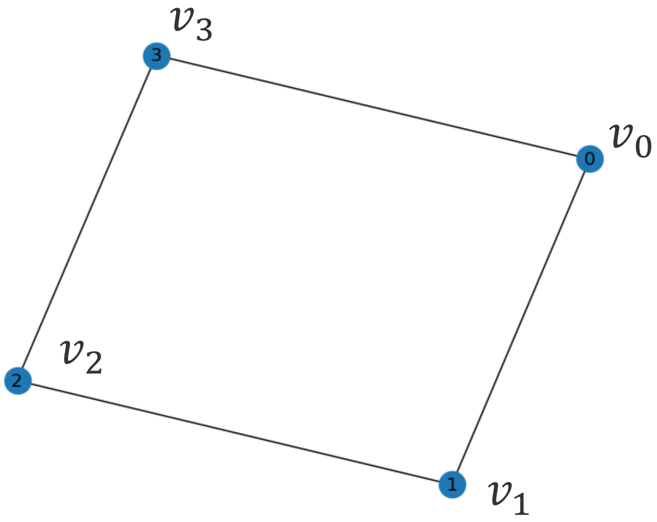
Quantum Approximate Optimization Algorithm Introduction

- MaxCut cost function
  1. Assign variable $v_i \in \{0,1\}$ to each vertex, that defines whether the vertex belongs to 0-partition or 1-partition

  2. Define a cost function, which encodes the solution to the problem

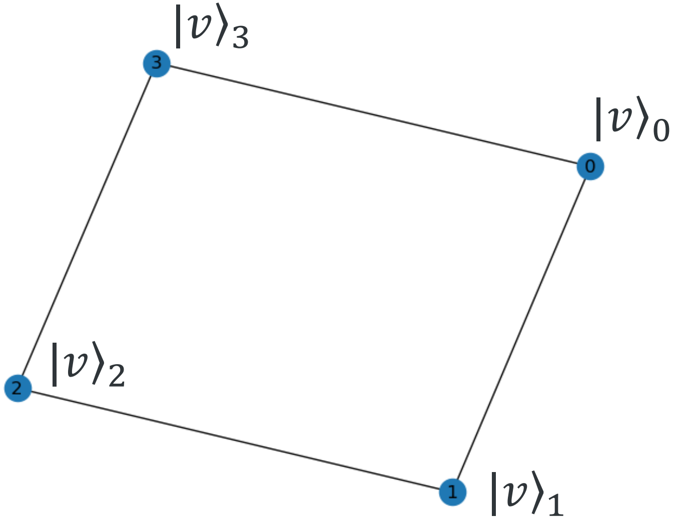
$$C = \underbrace{v_0 \wedge v_1}_{\substack{0 \text{ if } v_0 = v_1 \\ 1 \text{ if } v_0 \neq v_1}} + v_1 \wedge v_2 + v_2 \wedge v_3 + v_0 \wedge v_3$$

  3. Next, we move the cost function to energy landscape



Quantum Approximate Optimization Algorithm Introduction

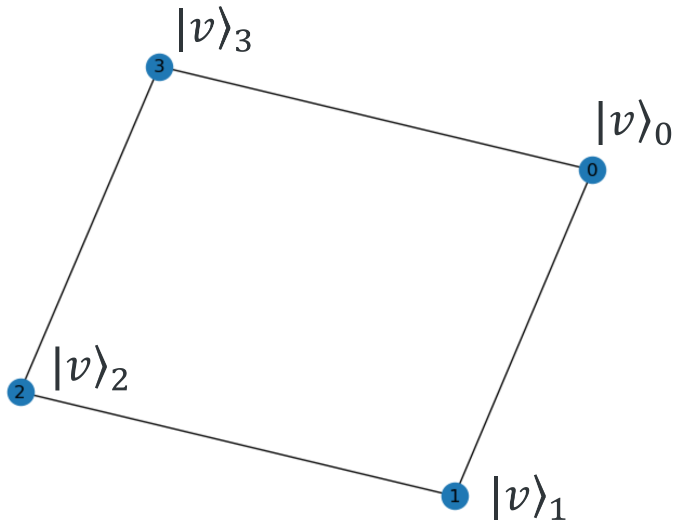
- Hamiltonian: (Quantum cost function)
  1. Now every vertex is represented by a qubit $|v\rangle_i$ and the total state of system is $|v\rangle = |v_0v_1v_2v_3\rangle$



Quantum Approximate Optimization Algorithm Introduction

- Hamiltonian: (Quantum cost function)
  1. Now every vertex is represented by a qubit $|v\rangle_i$ and the total state of system is $|v\rangle = |v_0 v_1 v_2 v_3\rangle$
  2. Energy of the system is governed by its Hamiltonian H. The expectation value should match the classical cost function:

$$\langle v|H|v\rangle = v_0 \wedge v_1 + v_1 \wedge v_2 + v_2 \wedge v_3 + v_0 \wedge v_3$$



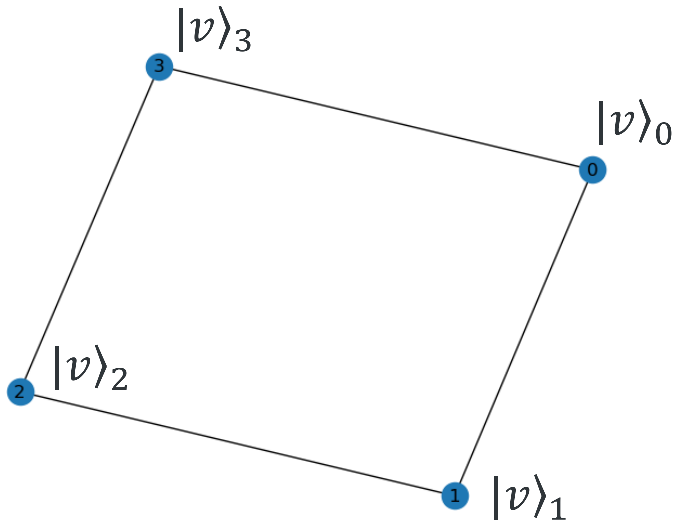
Quantum Approximate Optimization Algorithm Introduction

- Hamiltonian: (Quantum cost function)
  1. Now every vertex is represented by a qubit $|v\rangle_i$ and the total state of system is $|v\rangle = |v_0 v_1 v_2 v_3\rangle$
  2. Energy of the system is governed by its Hamiltonian H. The expectation value should match the classical cost function:

$$\langle v|H|v\rangle = v_0 \wedge v_1 + v_1 \wedge v_2 + v_2 \wedge v_3 + v_0 \wedge v_3$$

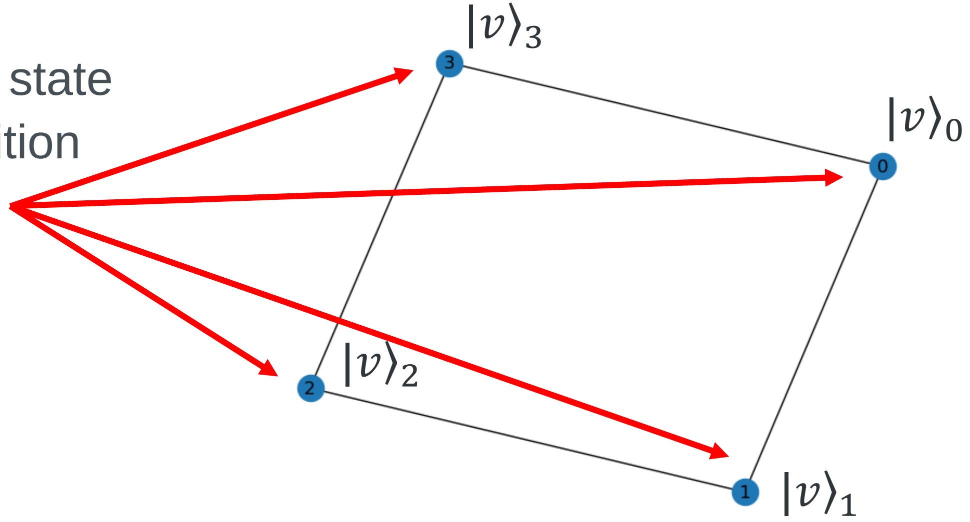
  3. By using Pauli Z-operations: $Z_i|0\rangle_i = |0\rangle_i$ and $Z_i|1\rangle_i = -|1\rangle_i$, we can find the matching expression:

$$H = \frac{1 - Z_0 Z_1}{2} + \frac{1 - Z_1 Z_2}{2} + \frac{1 - Z_2 Z_3}{2} + \frac{1 - Z_0 Z_3}{2}$$



Quantum Approximate Optimization Algorithm
Introduction

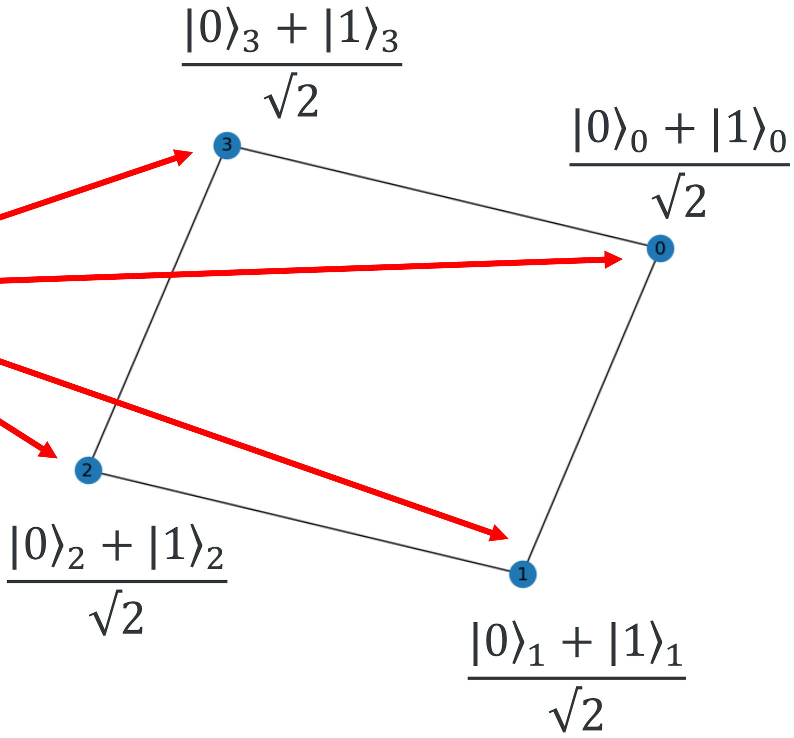
- When the ansatz that minimizes the energy is found, the most measured state for that ansatz holds tells which partition gives the MaxCut: $|v\rangle = |v_0 v_1 v_2 v_3\rangle$



Quantum Approximate Optimization Algorithm
Introduction

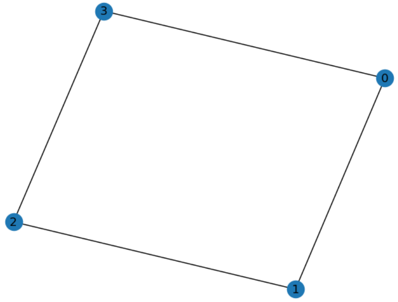
- When the ansatz that minimizes the energy is found, the most measured state for that ansatz holds tells which partition gives the MaxCut: $|v\rangle = |v_0 v_1 v_2 v_3\rangle$

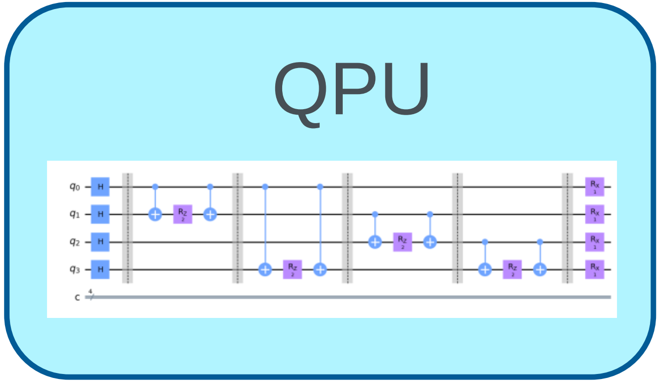
- Quantum computers can bring qubits to superposition to simultaneously evaluate every possible combination:
  - $|0000\rangle, |0001\rangle, \ldots, |1110\rangle, |1111\rangle$

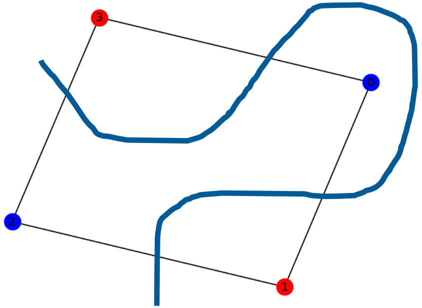


Quantum Approximate Optimization Algorithm Introduction

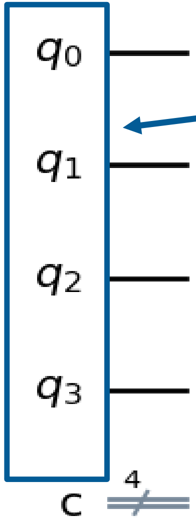
- Implementing MaxCut QAOA on a quantum computer







Quantum Approximate Optimization Algorithm Introduction

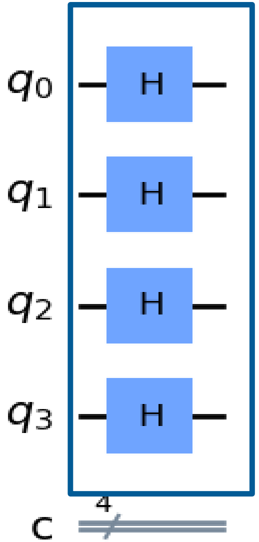


- Allocate one qubit for each vertex in the graph
- Every qubit starts at the ground state

$$|\psi\rangle = |0\rangle_0|0\rangle_1|0\rangle_2|0\rangle_3$$

$$= |0000\rangle$$

Quantum Approximate Optimization Algorithm Introduction



- Apply Hadamard gate to each qubit to create superposition of all possible solutions

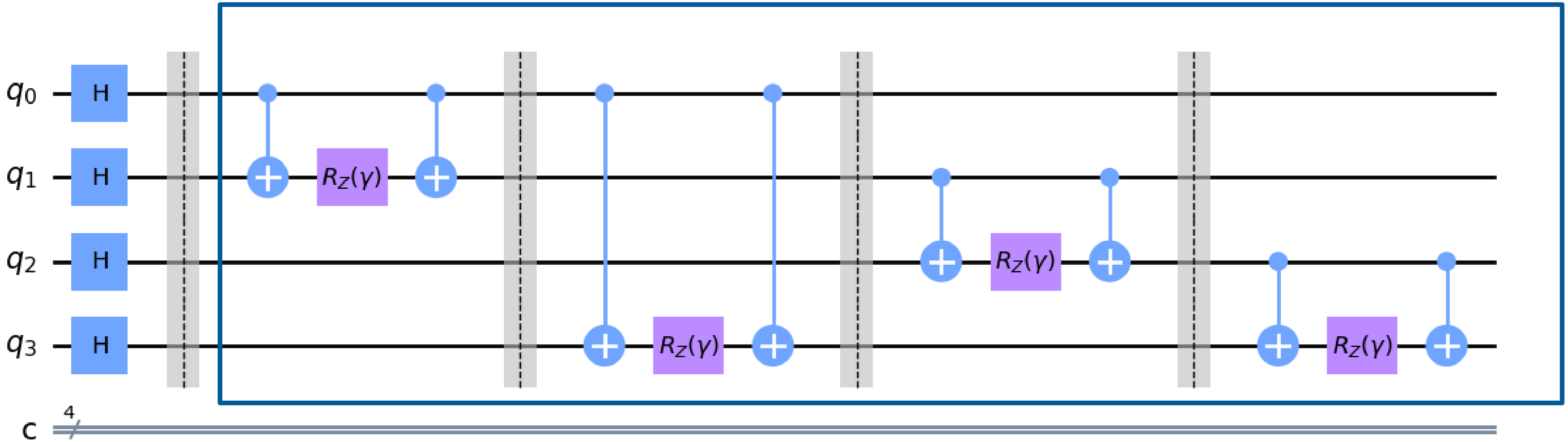
$$|\psi\rangle = \frac{|0\rangle_0 + |1\rangle_0}{\sqrt{2}} \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} \frac{|0\rangle_2 + |1\rangle_2}{\sqrt{2}} \frac{|0\rangle_3 + |1\rangle_3}{\sqrt{2}}$$

$$= \frac{|0000\rangle + |0001\rangle + \ldots + |1110\rangle + |1111\rangle}{4}$$

Quantum Approximate Optimization Algorithm
Introduction

Operate the system with our Hamiltonian

$$U(\gamma)_C = \prod_{(j,k)\in \text{edges}} e^{-i\gamma C_{jk}}, \qquad \text{where } C_{j,k} = \frac{1 - Z_j Z_k}{2}$$

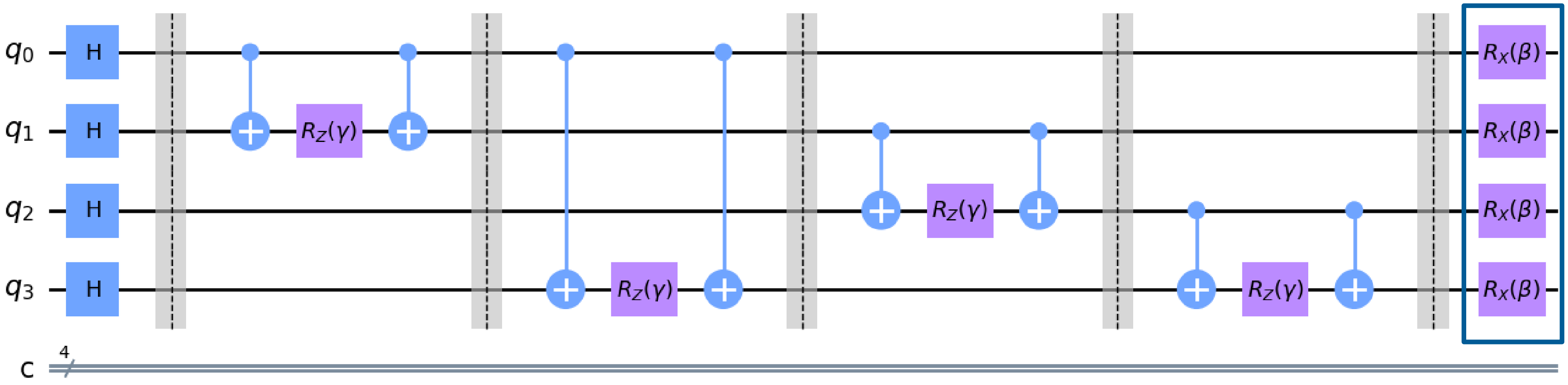


Quantum Approximate Optimization Algorithm Introduction

Apply mixer Hamiltonian

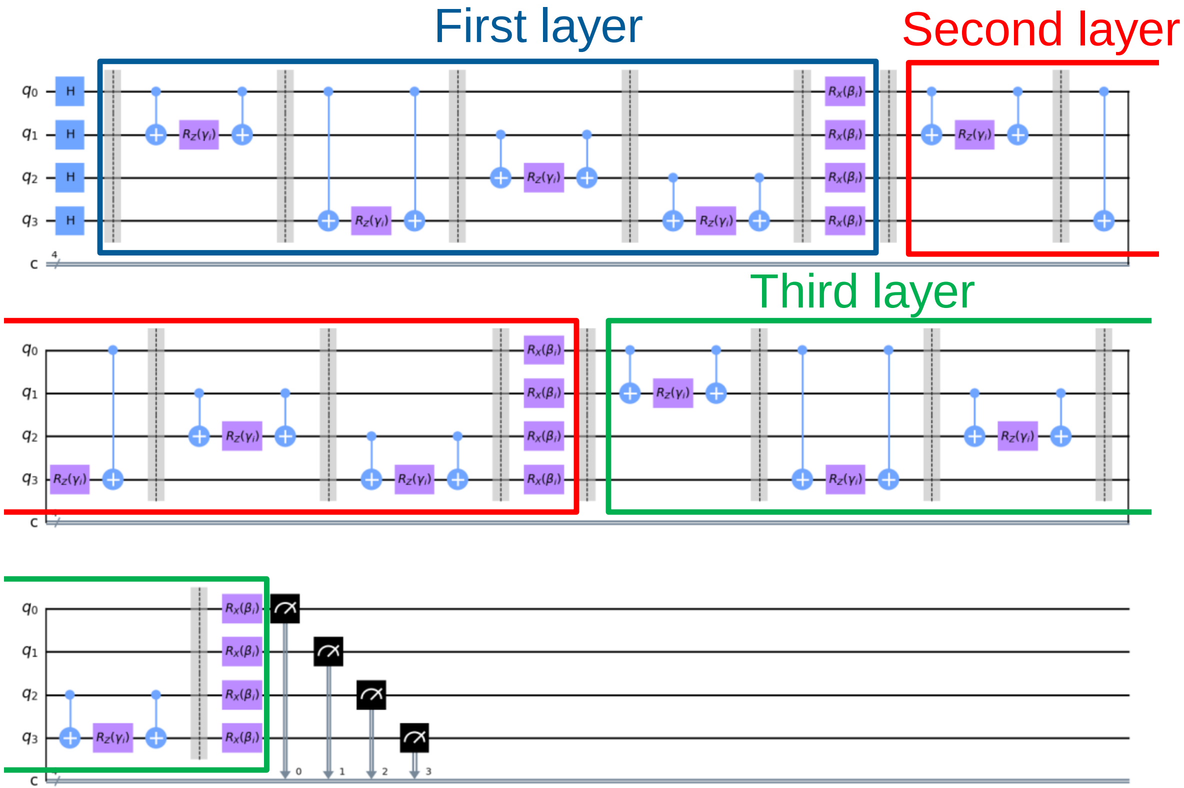
(in case the system gets 'stuck 'in its eigenstate)

$$U_B(\beta) = \prod_{j \in \text{vertices}} R_x(\beta)_j$$



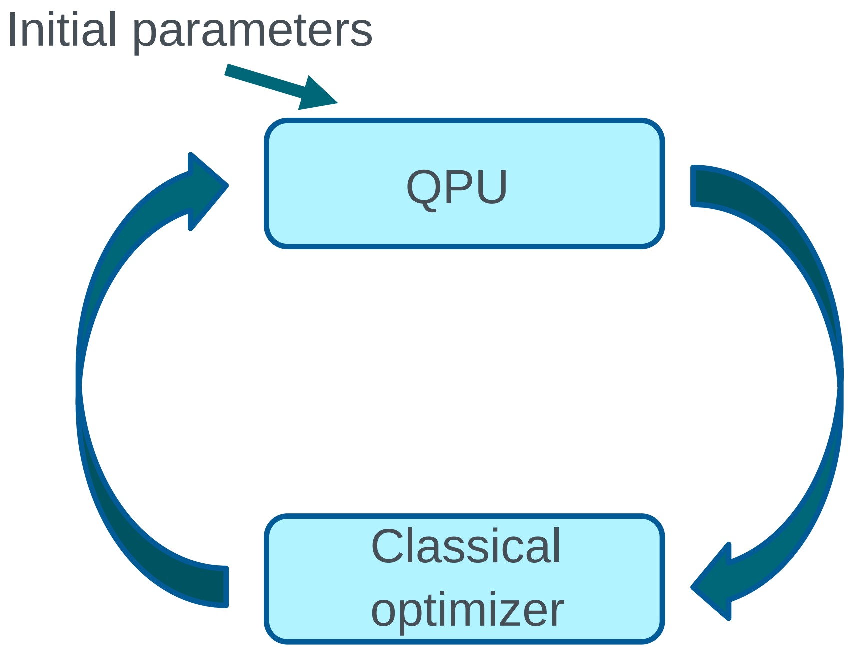
Quantum Approximate Optimization Algorithm Introduction

- Apply multiple layers of $U_C(\gamma)$ and $U_B(\beta)$ with the lists of parameters $[\gamma_0, \gamma_1, \ldots]$ and $[\beta_0, \beta_1, \ldots]$.
- Then measure all qubits



Quantum Approximate Optimization Algorithm Introduction

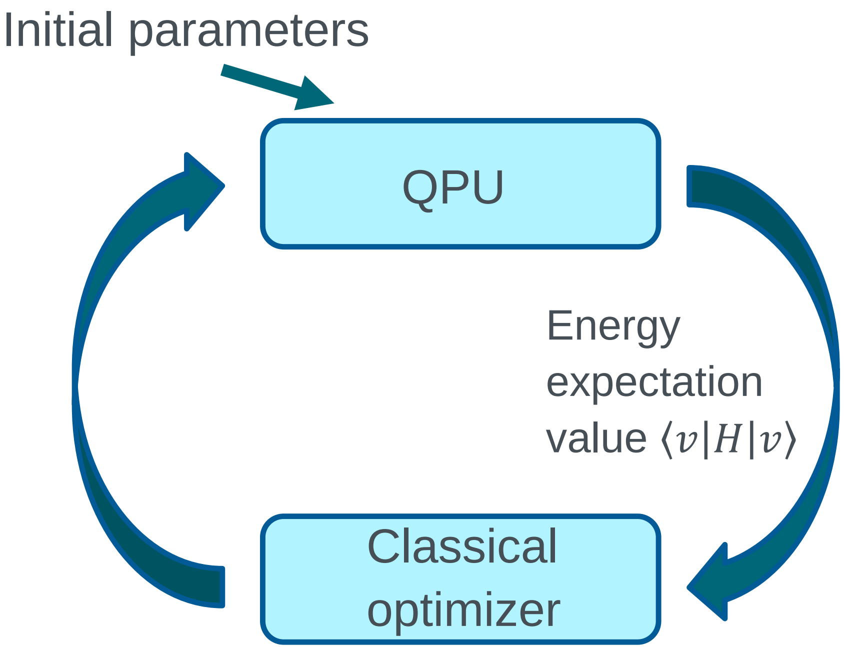
- Iterate to find to the correct solution
    1. Guess the quantum ansatz (initial parameters γ and β)



Quantum Approximate Optimization Algorithm Introduction

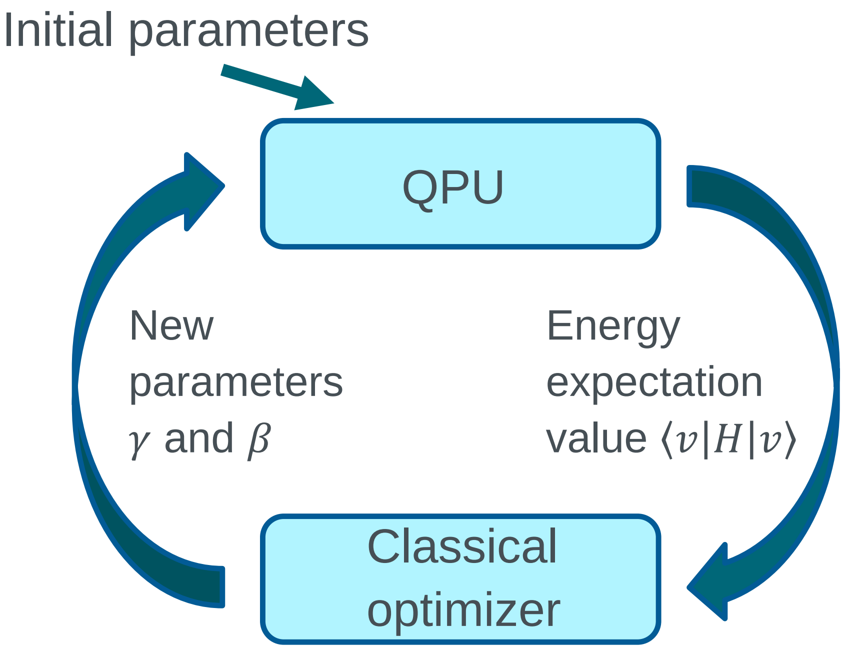
- Iterate to find to the correct solution
  1. Guess the quantum ansatz (initial parameters γ and β)

  2. Run the QAOA circuit and calculate the Hamiltonian expectation value (energy)



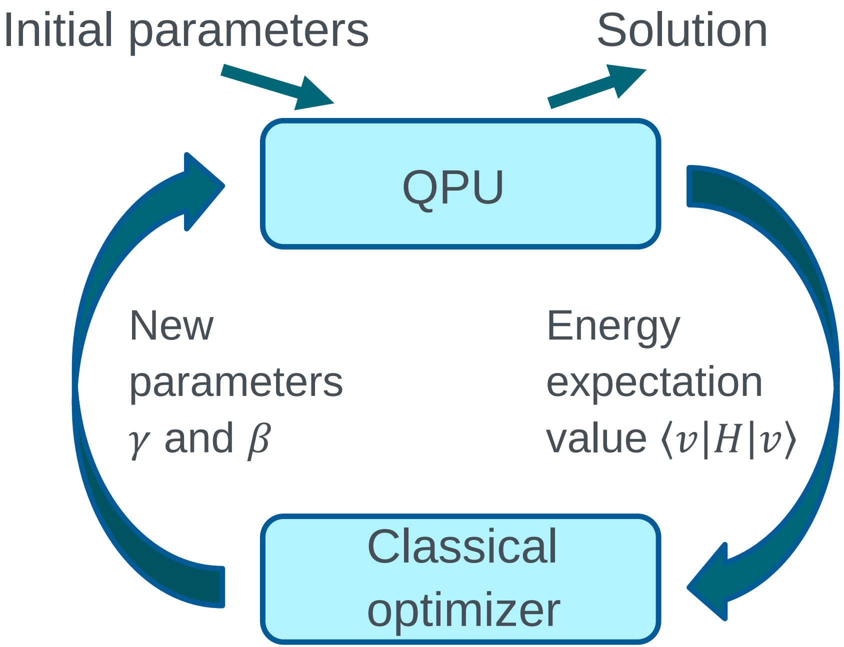
Quantum Approximate Optimization Algorithm Introduction

- Iterate to find to the correct solution
  1. Guess the quantum ansatz (initial parameters γ and β)
  2. Run the QAOA circuit and calculate the Hamiltonian expectation value (energy)
  3. Use classical optimizer to find the parameters that minimize the energy



Quantum Approximate Optimization Algorithm Introduction

- Iterate to find to the correct solution
  1. Guess the quantum ansatz (initial parameters γ and β)
  2. Run the QAOA circuit and calculate the Hamiltonian expectation value (energy)
  3. Use classical optimizer to find the parameters that minimize the energy
  4. Run the QAOA circuit with the optimal parameters; The most probable measured state gives the solution to MaxCut



Quantum Approximate Optimization Algorithm Introduction

- In summary: QAOA is a NISQ-era algorithm for optimization problems

Quantum Approximate Optimization Algorithm Introduction

- In summary: QAOA is a NISQ-era algorithm for optimization problems
- QAOA can solve a variety of optimization problems, which have many practical applications, i.e.
  - Chemistry/Physics: Hamiltonian simulations
  - Finance: Portfolio optimization
  - Machine learning: Training ML models

Quantum Approximate Optimization Algorithm Introduction

- In summary: QAOA is a NISQ-era algorithm for optimization problems
- QAOA can solve a variety of optimization problems, which have many practical applications, i.e.
  - Chemistry/Physics: Hamiltonian simulations
  - Finance: Portfolio optimization
  - Machine learning: Training ML models
- MaxCut itself has use cases in physics (Ising model) and machine learning (binary classification)

Quantum Approximate Optimization Algorithm Introduction

- In summary: QAOA is a NISQ-era algorithm for optimization problems
- QAOA can solve a variety of optimization problems, which have many practical applications, i.e.
  - Chemistry/Physics: Hamiltonian simulations
  - Finance: Portfolio optimization
  - Machine learning: Training ML models
- MaxCut itself has use cases in physics (Ising model) and machine learning (binary classification)
- Potentially faster than classical algorithms
  - Performs well in specific problems
  - Sometimes slowed down by parameters optimization and bad ansatzes