MARIST

School of Computer Science and Mathematics

Implementation of Quantum Algorithms and Optimization to Isolate DDoS Hive Plot Attacks

Meghan O’Loughlin

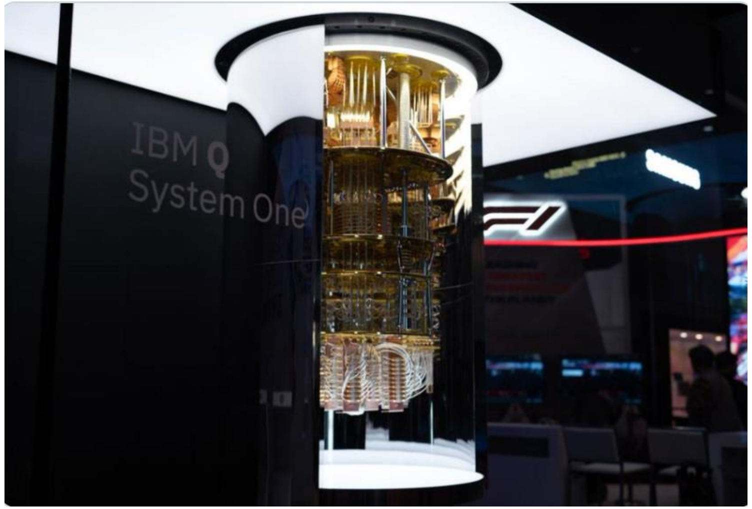
Undergraduate Student, Marist College

Casimer DeCusatis, Ph.D.

Associate Professor, Marist College, Poughkeepsie, NY
IBM Distinguished Engineer Emeritus

Agenda

- Conventional and Quantum Computing
- IBM Quantum Experience
- Max Cut Algorithm
- QAOA
- Results
- Future Work

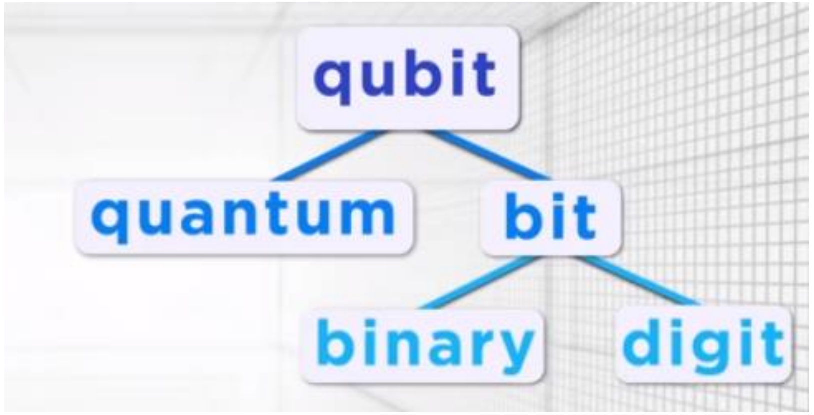


Limits of Conventional Computing

- What kinds of problems can't be solved by classical computers?
  - **Optimization** (best way to seat 10 people at a dinner table, or selecting the "best" route for thousands of cars traveling through a large city)
  - Simulating the real world (chemistry limited to a few atoms)
  - Problems with exponential speedup (chess board example)
  - Reversible computations (quantum computers should use far less energy)

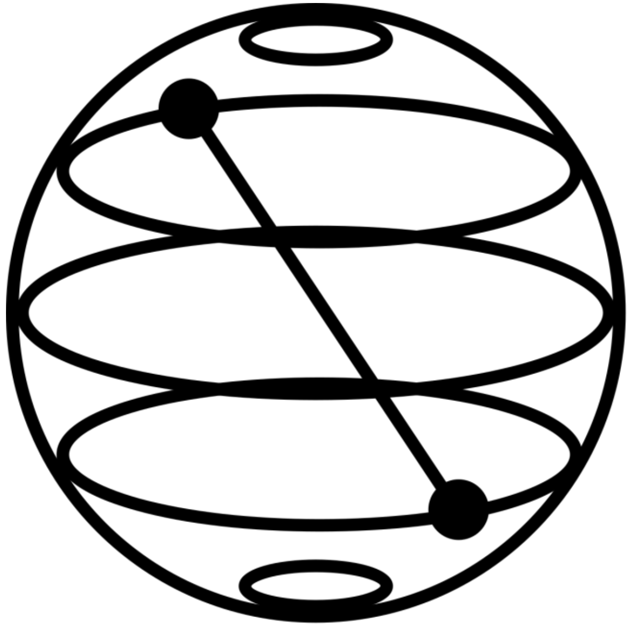
What is Quantum Computing?

- Classical Computing uses bits
- Quantum Computing uses ***qubits***
  - Superposition between 0 and 1
  - Can only be measured when it is collapsed
- A special type of co-processor that allows us to represent and manipulate data in different ways

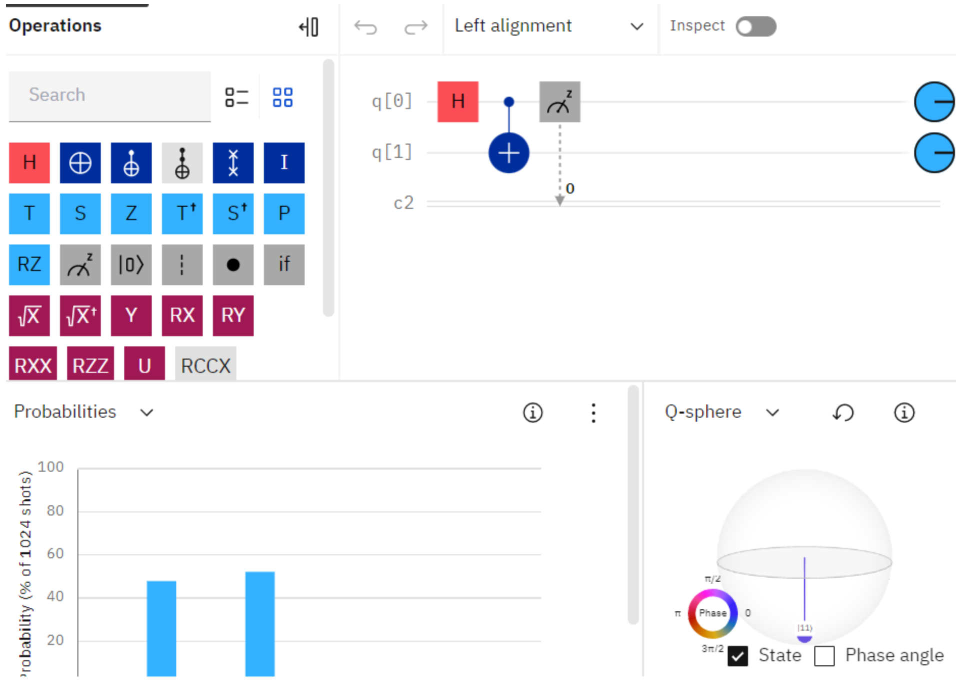


IBM Quantum Experience

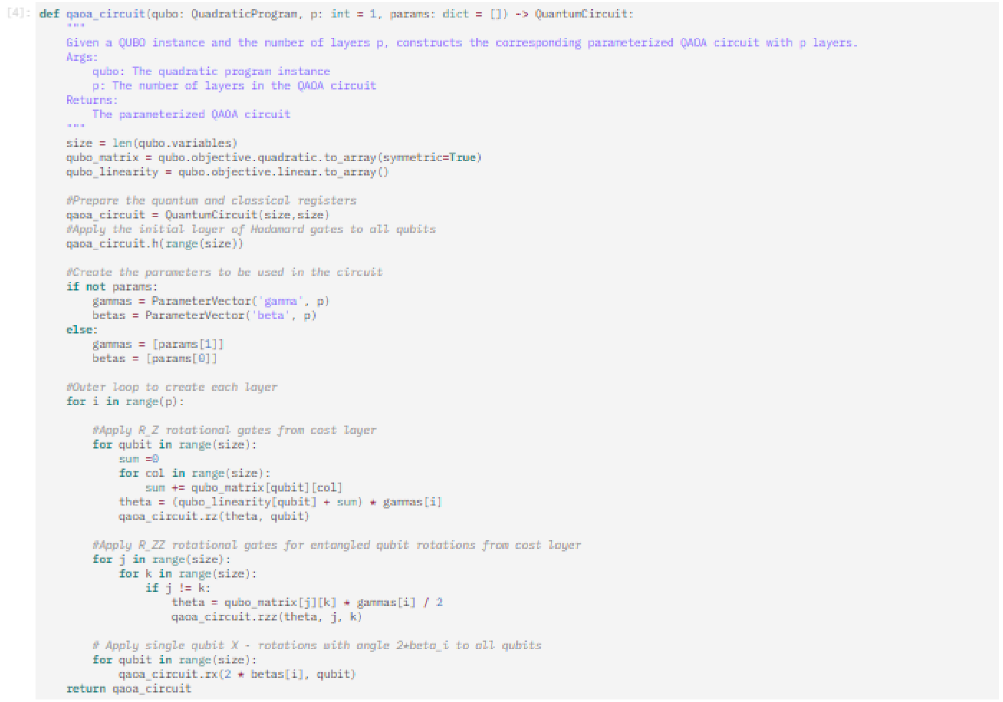
- IBM Q System One currently use ultra pure silicon atoms, shielded from ambient electromagnetic radiation and isolated from thermal noise
- Held at 0.015 Kelvin (almost the temperature of outer space)
- Qiskit programming language
- Currently limited to about 10 qubits, IBM plans to reach 1,000 qubits by 2023



IBM Quantum Experience



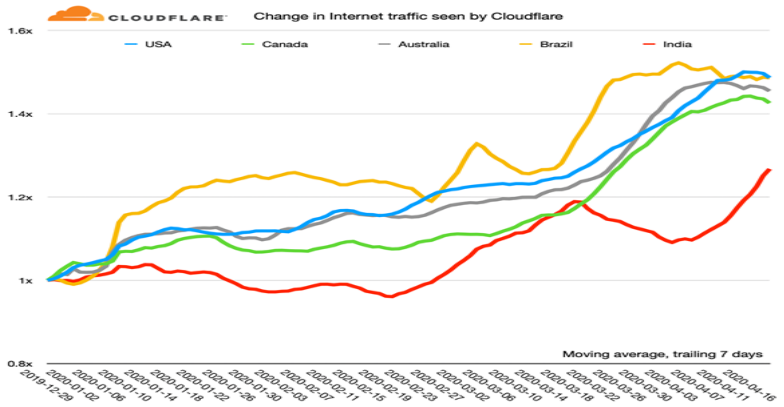
IBM Quantum Experience



```
[4]: def qaoa_circuit(qubo: QuadraticProgram, p: int = 1, params: dict = []) -> QuantumCircuit:
         """
         Given a QUBO instance and the number of layers p, constructs the corresponding parameterized QAOA circuit with p layers.
         Args:
             qubo: The quadratic program instance
             p: The number of layers in the QAOA circuit
         Returns:
             The parameterized QAOA circuit
         """
         size = len(qubo.variables)
         qubo_matrix = qubo.objective.quadratic.to_array(symmetric=True)
         qubo_linearity = qubo.objective.linear.to_array()

         #Prepare the quantum and classical registers
         qaoa_circuit = QuantumCircuit(size,size)
         #Apply the initial layer of Hadamard gates to all qubits
         qaoa_circuit.h(range(size))

         #Create the parameters to be used in the circuit
         if not params:
             gammas = ParameterVector('gamma', p)
             betas = ParameterVector('beta', p)
         else:
             gammas = [params[1]]
             betas = [params[0]]

         #Outer loop to create each layer
         for i in range(p):

             #Apply R_Z rotational gates from cost layer
             for qubit in range(size):
                 sum =0
                 for col in range(size):
                     sum += qubo_matrix[qubit][col]
                 theta = (qubo_linearity[qubit] + sum) * gammas[i]
                 qaoa_circuit.rz(theta, qubit)

             #Apply R_ZZ rotational gates for entangled qubit rotations from cost layer
             for j in range(size):
                 for k in range(size):
                     if j != k:
                         theta = qubo_matrix[j][k] * gammas[i] / 2
                         qaoa_circuit.rzz(theta, j, k)

             # Apply single qubit X - rotations with angle 2*beta_i to all qubits
             for qubit in range(size):
                 qaoa_circuit.rx(2 * betas[i], qubit)
         return qaoa_circuit
```

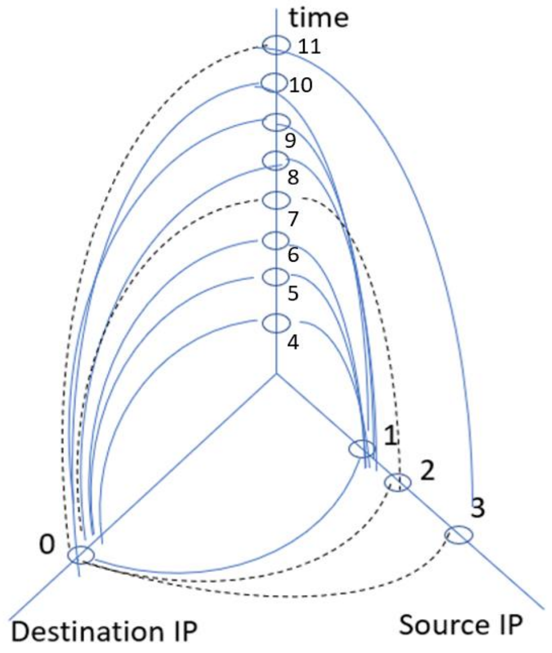
Cybersecurity DoS/DDoS Attack Problem

- Denial of Service (DoS) and Distributed DoS attacks flood a computer with network connection requests, forcing the computer offline
- During 2020 the average DDoS attack size increased over 540%, with the maximum attack size exceeding a terabit/second
- Potential impact of a cyberattack during the pandemic is increased in networks already strained for capacity
  - During 2020 ISP traffic grew over 30%
  - DDoS attacks are up 20% and growing



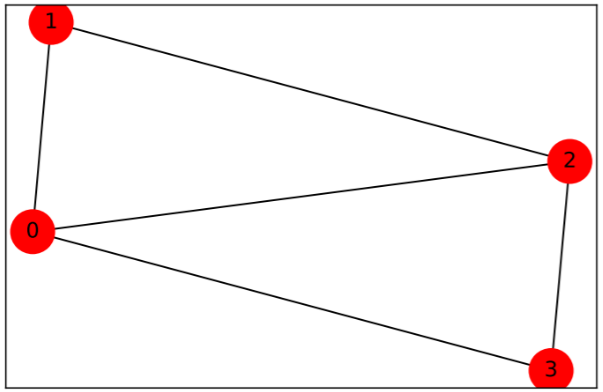
Marist Honeynet Data Sets

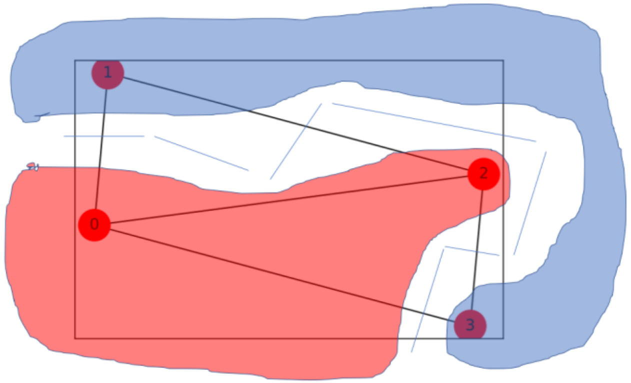
- Marist students previously developed a network of honeypots to collect and analyze various cybersecurity attack patterns
  - Collected a data set of 2,000 samples showing normal traffic and DDoS attacks every 15 seconds
- **Blocking a DDoS attack requires removing as many edges as possible between the botnet and target, while retaining edges not connected to the botnet. We need to rapidly divide the graph into attack and non-attack portions.**



Max/Min Cut Problem

- Given a graph with N nodes and weighted connections, how can we cut the graph into 2 parts that maximizes the value of the edges we cut?
  - Each node is assigned to either the first or second half of the cut



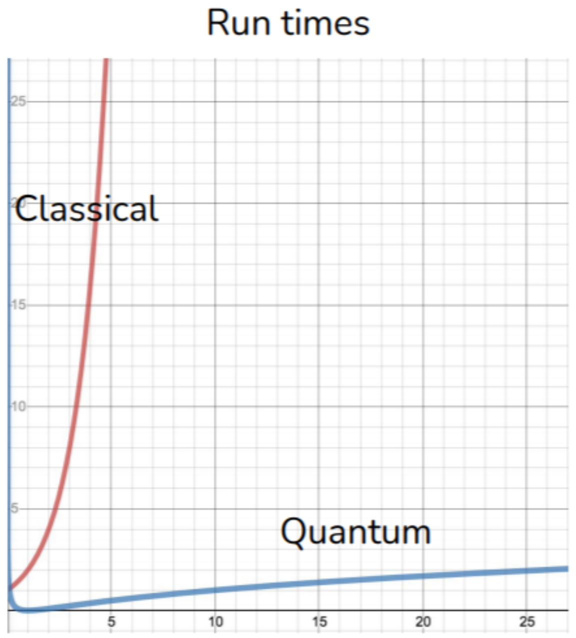


Classical vs Quantum Approach to Max/Min Cut

Classical algorithms need to exhaustively try all the possible binary assignments for each node and check the weight of the cut associated with each graph partition.
This is an NP-hard problem.

Quantum algorithms do not have a provable run time but appear to perform very well (polynomial run times) in many practical cases.

There is no known classical algorithm that can solve this problem in polynomial time



Optimization and QAOA

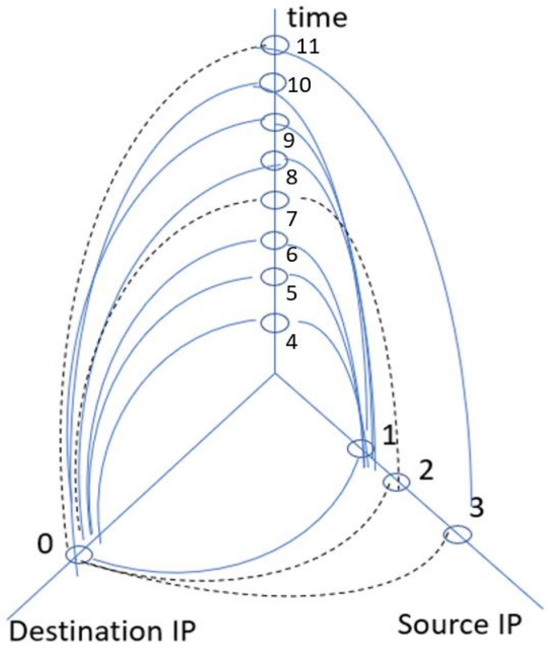
- Maximize or minimize a parameter
- Various quantum algorithms are applicable
  - QAOA (quantum approximate optimization algorithm)
  - VQE (variational quantum eigensolver)
  - QAE (quantum amplitude estimator)
- The QAOA algorithm is an example of combinatorial optimization, which attempts to find an optimal solution by maximizing a cost function.
  - Lowest energy state = optimal solution

$$\text{Energy}\,(|\psi>) = <\psi\,|\,H\,|\,\psi>$$

$$|\psi^{*}> = \text{argmin Energy}\,(\,|\psi>\,)$$

$$\text{Energy}\,(\theta*) = <\psi(\theta)\,|\,H\,|\,\psi(\theta)>$$

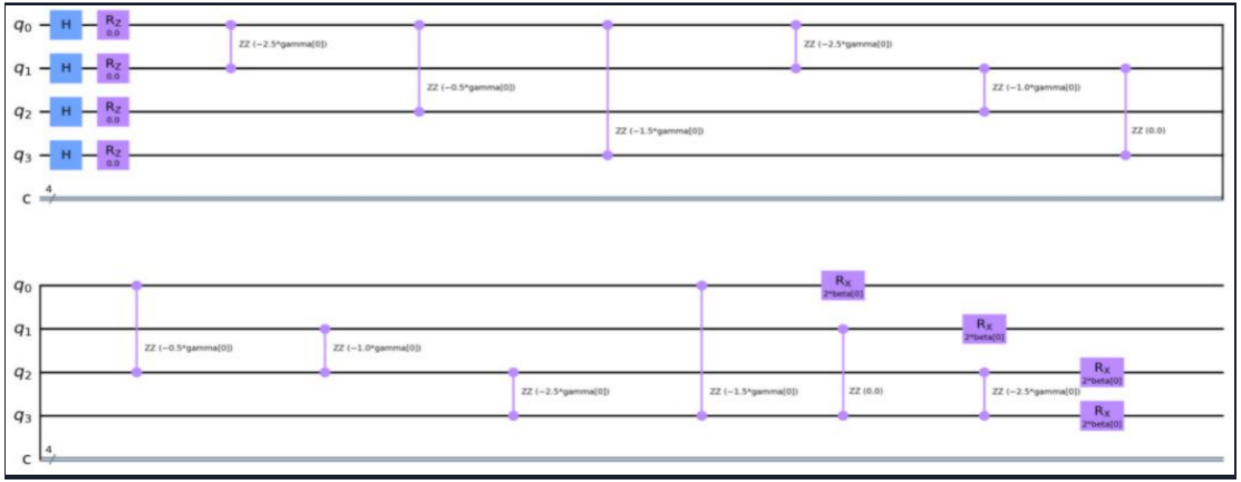
Input



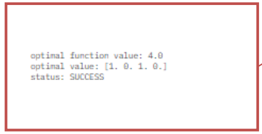
Node 0	[(0, 0, 0, 5, 5, 5, 5, 0, 5, 5),
Node 1	(0, 0, 1, 0, 0, 0, 0, 1, 0, 0),
Node 2	(0, 1, 0, 0, 0, 0, 0, 1, 0, 0),
Node 3	(5, 0, 0, 0, 5, 5, 5, 0, 5, 5),
Node 4	(5, 0, 0, 5, 0, 0, 0, 0, 0, 0),
Node 5	(5, 0, 0, 5, 0, 0, 0, 0, 0, 0),
Node 6	(5, 0, 0, 5, 0, 0, 0, 0, 0, 0),
Node 7	(0, 1, 1, 0, 0, 0, 0, 0, 0, 0),
Node 8	(5, 0, 0, 5, 0, 0, 0, 0, 0, 0),
Node 9	(5, 0, 0, 5, 0, 0, 0, 0, 0, 0)]

Algorithm

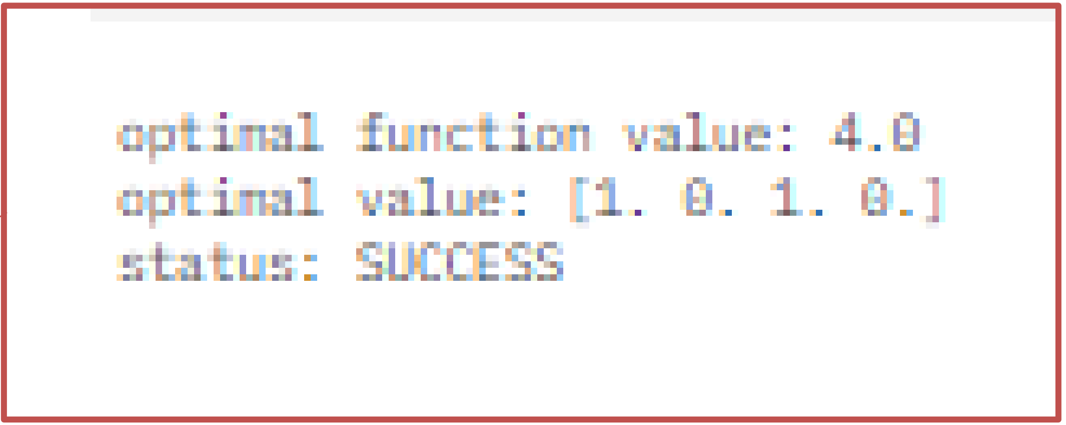
By running QAOA on a quantum computer, we calculate an energy value, which is passed to a classical optimizer that converges on an approximate solution.



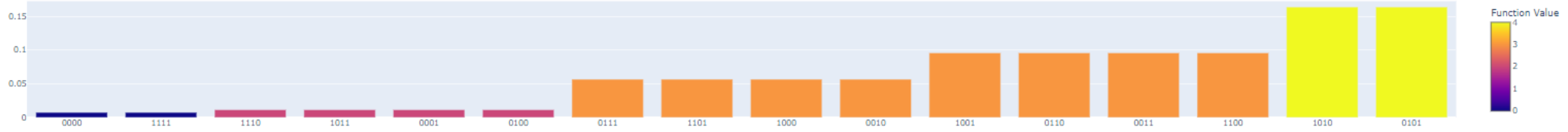
Output



```
optimal function value: 4.0
optimal value: [1. 0. 1. 0.]
status: SUCCESS
```

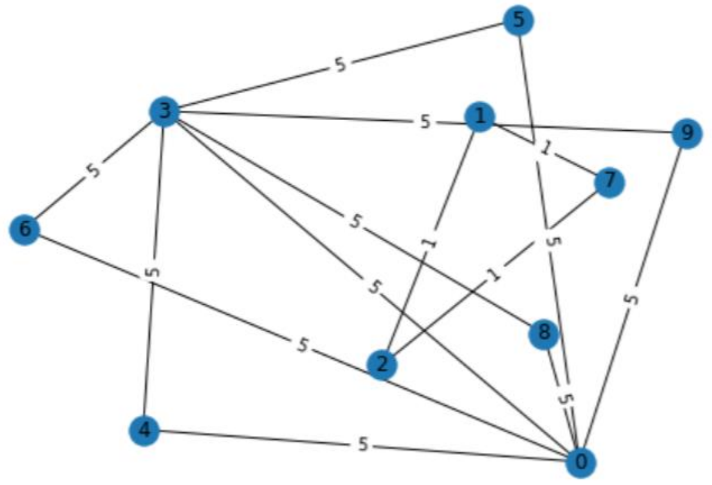


```
optimal function value: 4.0
optimal value: [1. 0. 1. 0.]
status: SUCCESS
```

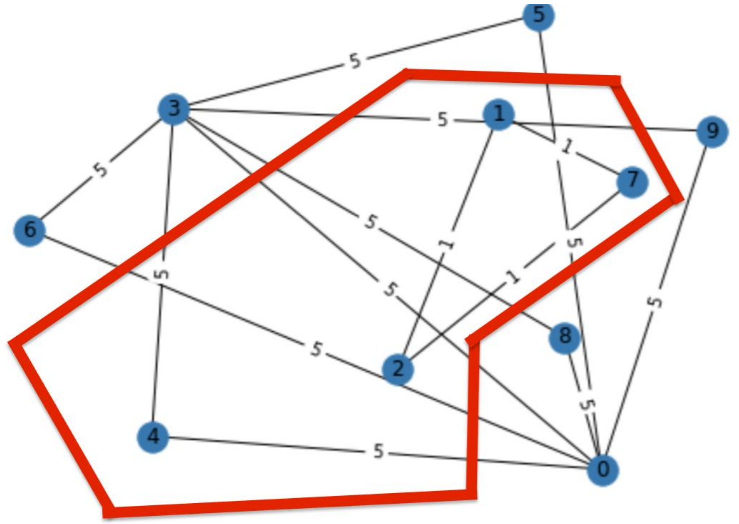


Experimental Results DDoS

Input

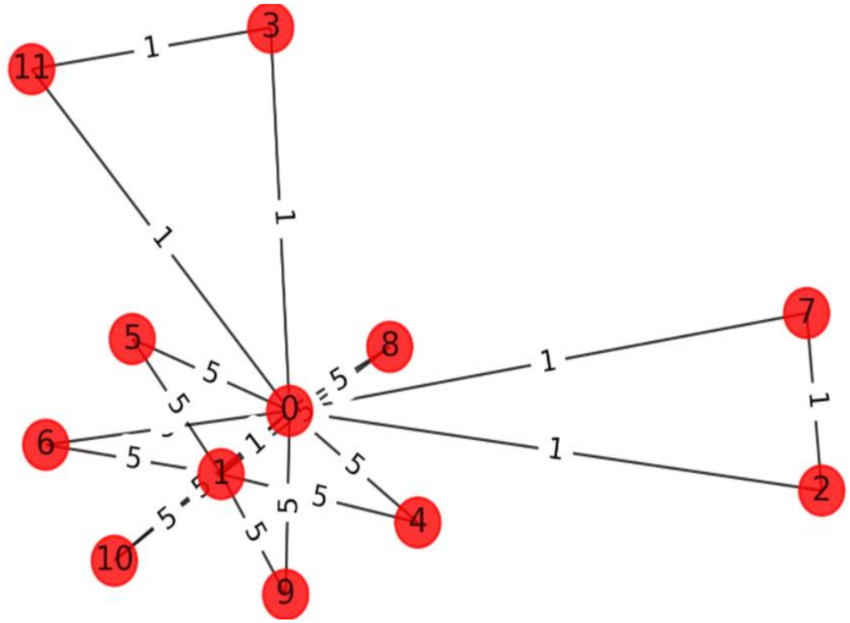


Max Cut Output

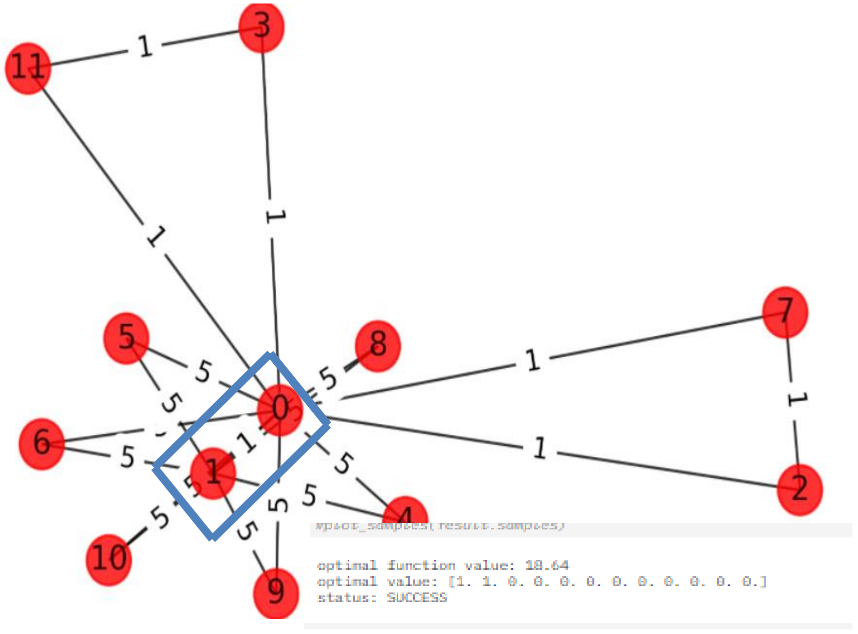


Experimental Results DoS

Input



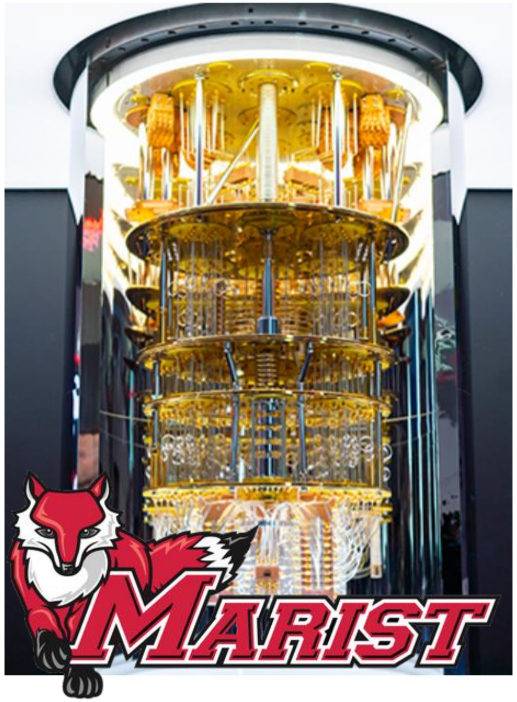
Max Cut Output



```
#plot_samples(result.samples)

optimal function value: 18.64
optimal value: [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
status: SUCCESS
```

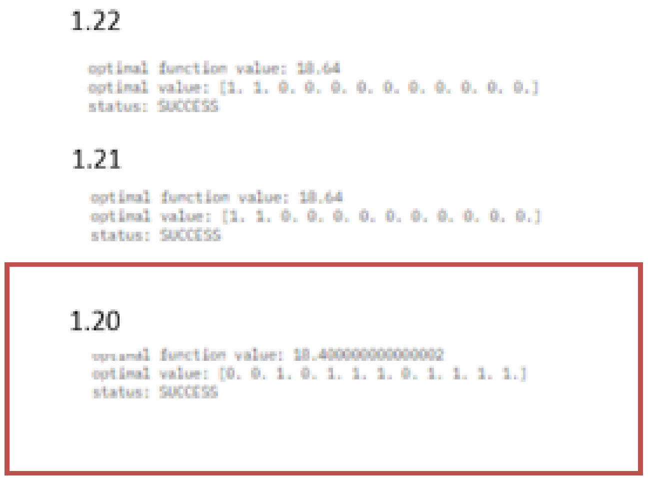
Conclusions

- Using superposition and entanglement, can potentially solve NP-hard problems
- **80-90% accuracy**
  - Potential to be improved with a larger number of qubits
- Use Quantum algorithms to parse a graph
  - Works for small graphs
  - Believe it will allow for better/faster partitioning
  - **Hand data over to SOC for analysis**



Future Work and Challenges

- Proof of concept
- Do some more work to validate and scale
- IBM Quantum Experience limited to 5-8 qubits for public use
  - Spin up VM to simulate on larger scale
- Understand the weight problem



```
1.22

  optimal function value: 10.64
  optimal value: [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
  status: SUCCESS

1.21

  optimal function value: 10.64
  optimal value: [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
  status: SUCCESS

1.20
  optimal function value: 10.400000000000002
  optimal value: [0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1.]
  status: SUCCESS
```