# Quantum Approximate Optimization Algorithm (QAOA) and Quantum Annealing for Combinatorial Optimization

David E. Bernal, Baljit Singh, Prateek Singh, Mahmoud Al Ismail

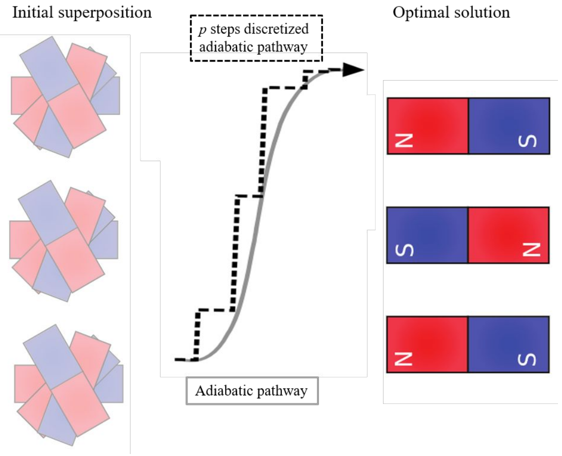
11-860 Quantum Computing Lab - Fall 2020 - CMU

# Quantum Applications

- Quantum computers have shown to theoretically solve some problems better than classical computers
  - Factoring (Shor's algorithm)
  - Search (Grover's algorithm)
- Current State
  - With error correction playing a significant role, we need large number of qubits ( probably more than 1000s) to run the algorithm
  - Quantum computers can use upto 50-70 qubits for calculation
- Near-Term Quantum algorithms
  - Can run on small quantum computers
  - Require only a small number of qubits for calculation
  - Don't require extensive error correction

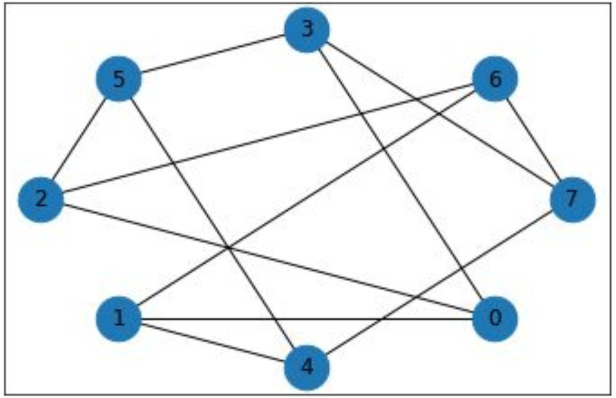
# Quantum Approximation Optimization Algorithm

- Proposed by Farhi et. al 2014
- Can be understood as discretized version of the Quantum Adiabatic Alg.
  - Also proposed by Fahri et. al 2001
  - Discretized Adiabatic Quantum Algorithm such that slow evolution replaced by series of rotations
- Properties:
  - **low depth**
  - can be implemented on near-term quantum computers
  - requires as many qubits as the size of the problem
  - more robust to errors[source]



# QAOA (cont'd)

- Farhi et. al applied this algorithm on MaxCut and provided an approximation ratio of 0.6942
- Naive algorithm that yields an approximation ratio of 0.66
- Evidently, QAOA did give an improvement!
- Since then, QAOA has been applied to different optimization problems:
  - MAX-2-SAT
  - TSP
  - Graph Coloring
  - Single Machine Scheduling



# Quantum Annealing (QA)

- Initially proposed as quantum counterpart of simulated annealing (SA)
- Practically implements Quantum Adiabatic Algorithm
- Properties:
  - **low depth**
  - can be implemented on near-term quantum computers
  - requires as many qubits as the size of the problem
  - more robust to errors[source]

# Combinatorial Optimization

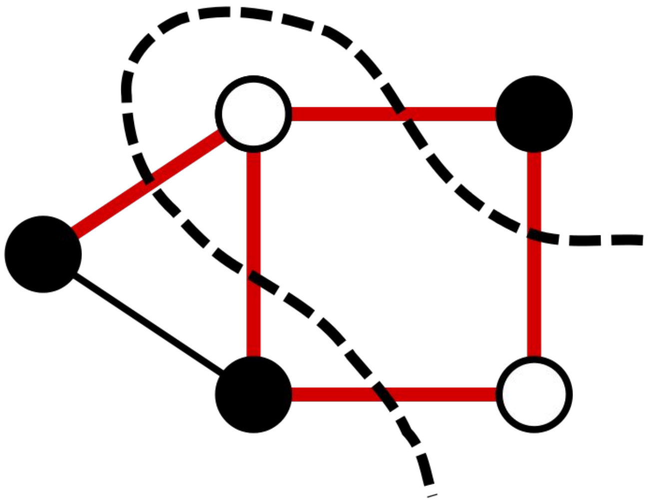
- QAOA was designed to provide approximate solutions for combinatorial optimization problems
- Combinatorial optimization is defined by $n$ bits and $m$ clauses
- We can define the objective function:

$$C(z) = \Sigma_{\alpha=1}^{m} C_\alpha(z)$$

- where $z = z_1 z_2 \ldots z_n$ is the bit string and $C_\alpha(x) = 1$ and $0$ otherwise

# MaxCut

- Given a graph $G = (V, E)$ with vertices $v \in V$ and edges $e_{j,i} \in E$, that map between two vertices in V.
- The maximum cut of the graph G divides the vertices into two disjoint subsets.
- The number of edges between the vertices from the two sets is maximized



# MaxCut - Formulation

Aim: Divide the vertices such that number of edges with endpoints in different sets is maximized

Given a graph G = (V,E) we can create a cut by assigning each vertex either +1 or -1

Let $f(i) = \begin{cases} +1, & \text{if vertex } i \text{ is in set } A \\ -1, & \text{otherwise} \end{cases}$

The cost function for a given edge between two vertices $k$ and $j$ would be

$$C = \Sigma_{\langle j,k \rangle} C_{\langle j,k \rangle}$$

$$C_{\langle j,k \rangle} = \frac{1}{2}(-f(j)f(k) + 1) = \begin{cases} 1, & \text{if edge } \langle jk \rangle \text{ is a cut,} \\ 0, & \text{otherwise} \end{cases}$$

# MaxCut - Formulation with QAOA

We can define $C$ as a diagonal operator on $|z\rangle$ (operates on the $2^n$ Hilbert space) where

$$C|z\rangle = \sum_{\langle jk \rangle} C_{\langle jk \rangle}(z)|z\rangle$$

where each vertex is mapped a single qubit.

Now, our objective becomes $C_{\langle jk \rangle}|z\rangle = \frac{1}{2}(-f_j \otimes f_k + I)\,|z\rangle$ such that:

$$C_{\langle jk \rangle}|z\rangle = \begin{cases} |z\rangle, & \text{if edge } \langle jk \rangle \text{ is a cut, that is } f_j \otimes f_k|z\rangle = \text{-I} \\ 0, & \text{if edge } \langle jk \rangle \text{ is not a cut, that is } f_j \otimes f_k|z\rangle = \text{I} \end{cases}$$

What gate would we could use to exhibit this behavior in a quantum computer?

We can use $\sigma^z$ (Pauli-Z)! Confirmation: $\sigma^z|1\rangle = -1$ and $\sigma^z|0\rangle = 1$

Thus, the operator becomes $C_{\langle jk \rangle} = \frac{1}{2}(-\sigma_j^z \otimes \sigma_k^z)$

# MaxCut - Formulation with QAOA

We can define two unitary rotations, as follows:

$$U\left(C,\gamma\right) = e^{-i\gamma C} = \prod_{\langle jk \rangle} U\left(C_{\langle jk \rangle}, \gamma\right)$$

$$U\left(C,\gamma\right) = e^{-i\gamma\frac{1}{2}\left(-\sigma_j^z \otimes \sigma_k^z + I\right)} = e^{-i\frac{-\gamma}{2}\left(-\sigma_j^z \otimes \sigma_k^z\right)} e^{-i\gamma\frac{1}{2}I}$$

and,

$$U\left(B,\beta\right) = e^{-i\beta B} = \prod_{j=1}^{n} U(B_j, \beta)$$

$$where,$$

$$U\left(B_j,\beta\right) = e^{-i\beta\sigma_j^x}$$

# MaxCut - Formulation with QAOA

We can define our initial state as a transformation under $U(B, \beta)$ and $U(C, \gamma)$ where

$$|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p) \ldots U(B, \beta_1)U(C, \gamma_1)|s\rangle$$

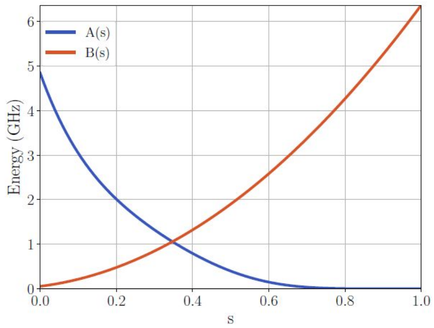
parameterised by $2p$ angles, $\gamma = \gamma_1, \ldots, \gamma_p$ and $\beta = \beta_1, \ldots, \beta_p$.

We can take the expectation, $F_p$, of $C$ within the above state

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

Then our goal in QAOA is to maximize the expectation w.r.t $\gamma, \beta$

# MaxCut - Formulation via Ising for QA




In Quantum Adiabatic Algorithm the Hamiltonian is defined as

$$H(s) = A(s)H_i + B(s)H_f$$

Where an initial Hamiltonian $H_i$ is evolved to a final Hamiltonian $H_f$ via some functions $A(s), B(s)$ of the adimensional time $s = t/T$

The initial Hamiltonian $H_i = \sum_{i=1}^{n} \sigma_i^x$ is decided such that the state is initialized at its minimum energy (superposition)

The final Hamiltonian can be tuned by matrix $J$ and vector $h$ in an Ising model formulated as

$$H_f = \sum_{(ij)\in E} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i\in V} h_i\sigma_i^z$$

# MaxCut - Formulation via Ising for QA

Using the same variables $f(i)$ for every vertex being on either side of the cut we define the Max-cut problem as

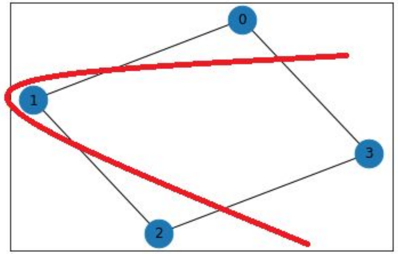
$$C_{max} = \max_{\mathbf{f}\in\{\mathbf{+1},\mathbf{-1}\}^{\mathbf{n}}} \frac{1}{2}\sum_{(j,k)\in E} w_{ij}(-f(j)f(k)+1)$$

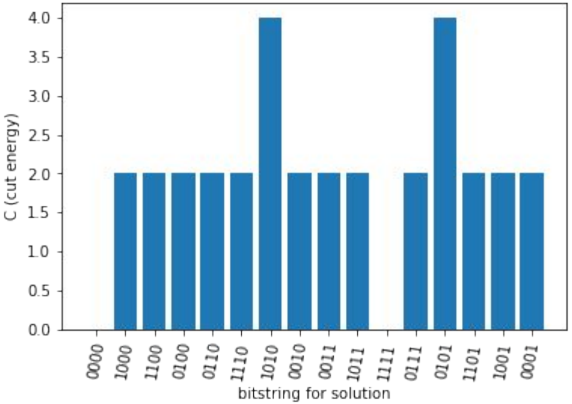
With $w_{jk}$ being the weight of edge $(jk)$.

Then by adding an offset of $\frac{1}{2}\sum_{(j,k)\in E} w_{jk}$, and setting $J_{jk} = \frac{1}{2}w_{jk}, h_j = 0$

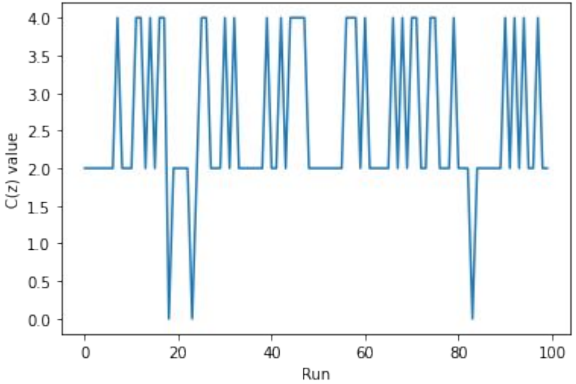
We map Max-cut to an Ising model.

# Results - Small degree-2 graph - Square

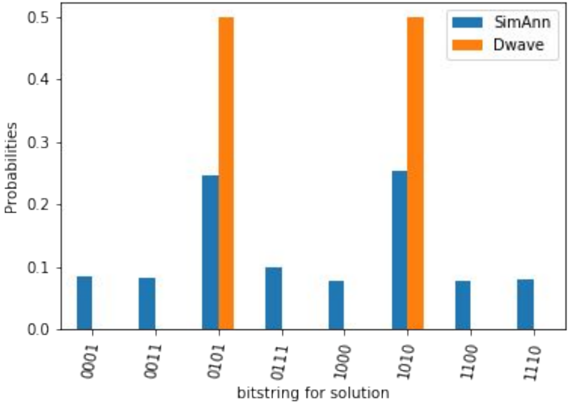



(a) Cut  values for all bitstrings



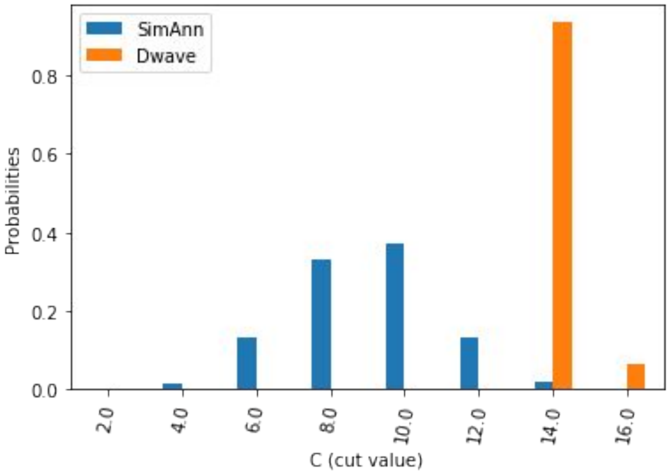

(b) QAOA for arbitrary angles

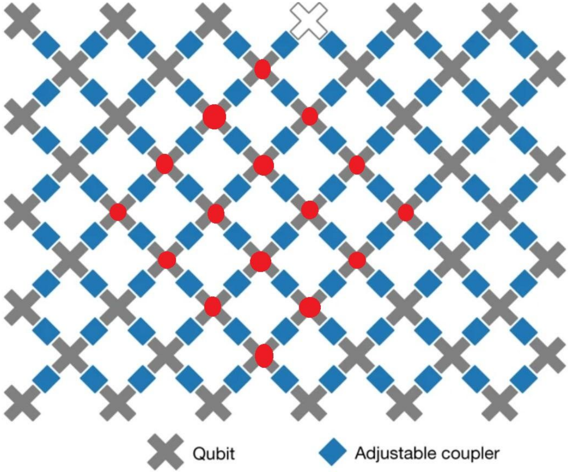


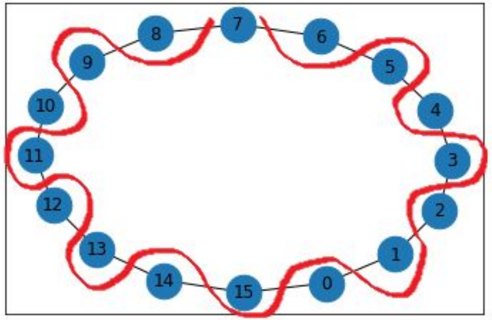

(c) Outcome probability for DWave

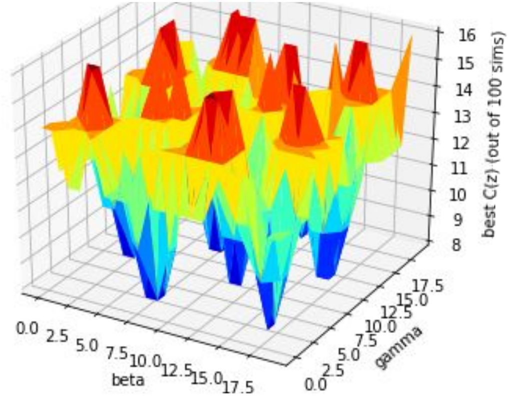
```
(0, 0): ───H───@──────────────────@──────────────────────────────────────────────X───Rz(-0.127π)───X───H───Rz(0.127π)───H───M('0')───
               │                  │                                              │                 │
(0, 1): ───H───X───Rz(-0.127π)───X───@──────────────────@────────────────────────┼─────────────────┼───H───Rz(0.127π)───H───M('1')───
                                     │                  │                        │                 │
(1, 0): ───H─────────────────────────X───Rz(-0.127π)───X───@──────────────────@──┼─────────────────┼───H───Rz(0.127π)───H───M('2')───
                                                           │                  │  │                 │
(1, 1): ───H───────────────────────────────────────────────X───Rz(-0.127π)───X───@─────────────────@───H───Rz(0.127π)───H───M('3')───
```

# Results - Degree-2 graph - Cycle



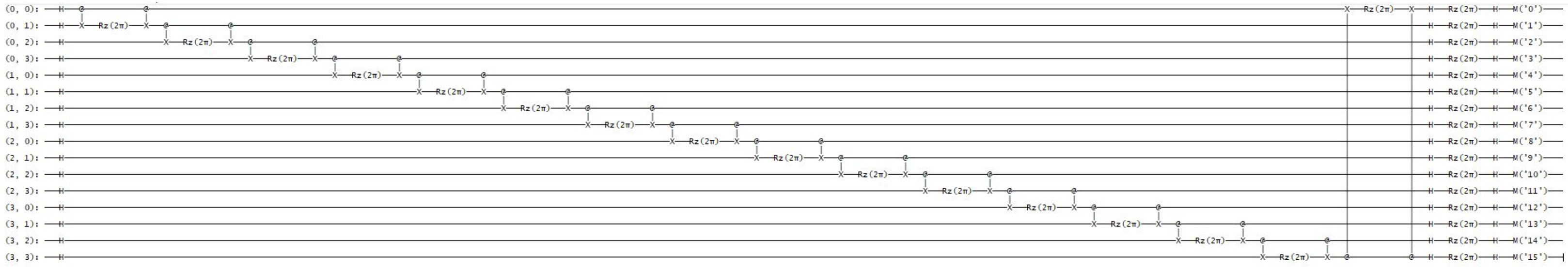



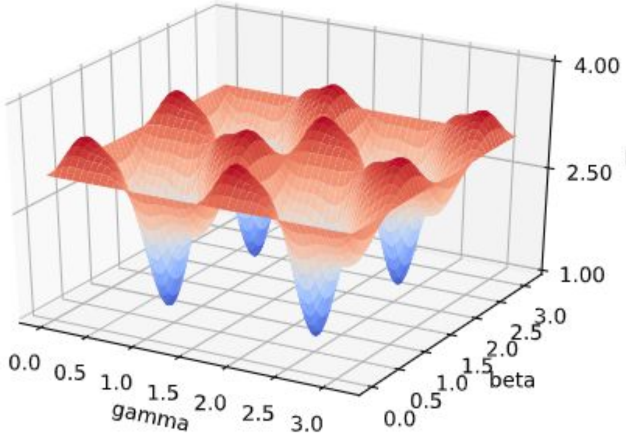

(a) Energy probability for DWave

(b) Google's Sycamore chip

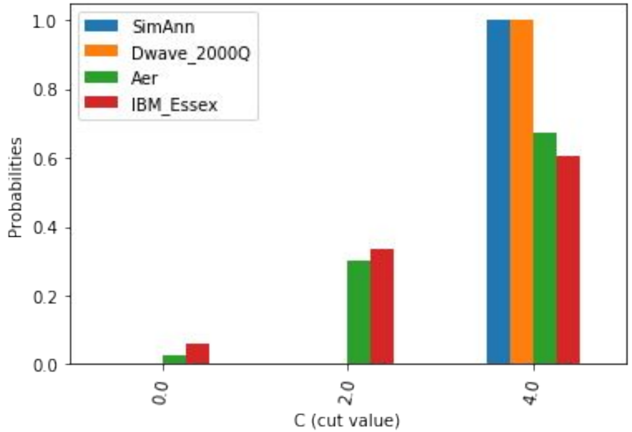
(c) Best found energy w.r.t angles



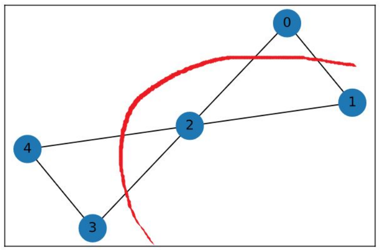
# Results - Butterfly graph

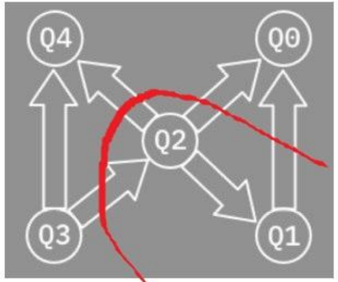


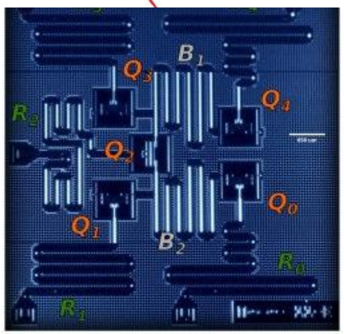

(a) Expected cost w.r.t. angles in QAOA

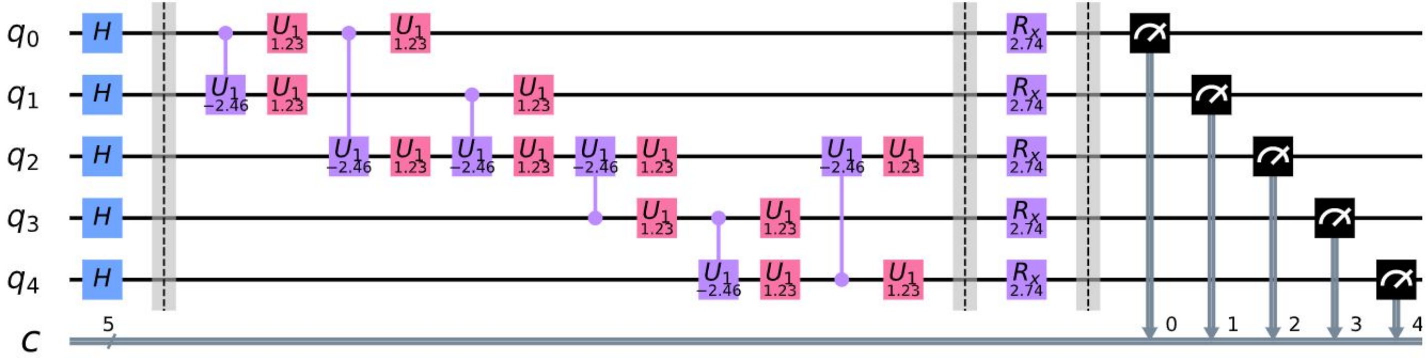



(b) Energy probability for DWave and IBM



# Discussion

- What has QAOA been through yet?
  - Was able to beat the best known classical algorithm for MaxCut solution approximation
  - Theorists developed a more efficient classical algorithm than the quantum algorithm [source]

- Is QAOA useful?
  - Sure! No harm in assuming
  - Useful but limited by the quantum hardware (gate infidelities)
  - The performance of QAOA depends on the the angles obtained through local search on a classical computer
  - Quantum Supremacy through QAOA on noisy intermediate-scale quantum computers [source]