How to run a near term Quantum Computer

Edward Farhi

Google / MIT

Quantum Computing

Conventional Computer Bits

In	0	1	1	1	0	1	1	0
Out	1	1	0	1	1	0	1	1

Quantum Computer Qubits

In	↑	↓	↓	↓	↑	↓	↓	↑
Out	↓	↓	↑	↓	↓	↑	↓	↓

Combinatorial Optimization

n bits   m clauses

Each clause depends on a few bits

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \qquad z_1 \ldots z_n = z$$

$$C_\alpha(z) = \begin{cases} 1 & \text{satisfies} \\ 0 & \text{does not} \end{cases}$$

$$C_{max} = \max_z C(z)$$

Want $\boxed{\dfrac{C(z)}{C_{max}}}$ big.

Quantum Algorithm Ingredients:

$$\underline{U(C,\gamma)} = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

$$B = \sum_{j=1}^{n} X_j \qquad X_j = \sigma_j^x$$

$$\underline{U(B,\beta)} = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

$$\underline{|s\rangle} = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle$$

All easy to construct!

For any integer $p \geq 1$ $\quad \underline{\gamma_1 \ldots \gamma_p = \vec{\gamma}} \quad \underline{\beta_1 \ldots \beta_p = \vec{\beta}}$

$$\underline{|\vec{\gamma}, \vec{\beta}\rangle} = U(B, \beta_p)\, U(C, \gamma_p) \cdots U(B, \beta_1)\, U(C, \gamma_1)\, |s\rangle$$

Required Circuit Depth at most $mp + p$.

$$F_p(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | C | \vec{\gamma}, \vec{\beta} \rangle$$

$$M_p = \max_{\vec{\gamma}, \vec{\beta}} F_p(\vec{\gamma}, \vec{\beta})$$

$$\boxed{M_p \geq M_{p-1}}$$

Can show $\boxed{\lim_{p \to \infty} M_p = C_{max}}$

Quantum Algorithm with Angle Search

Fix p. Start with angles $(\vec{\gamma}, \vec{\beta})$

Use the Quantum Computer to make

$$|\vec{\gamma}, \vec{\beta}\rangle$$

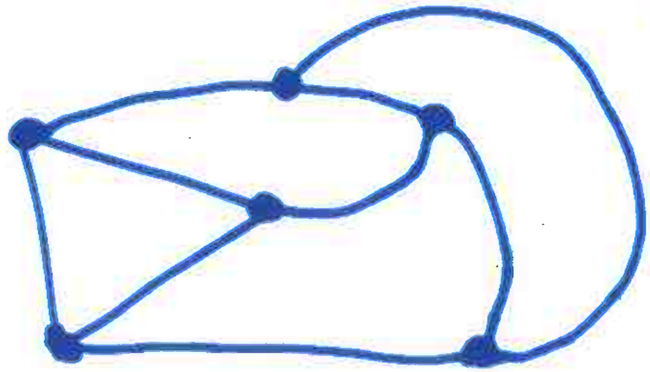
Measure to get a string z and $C(z)$.

Repeat with same angles to get a good estimate of $F_p(\vec{\gamma}, \vec{\beta})$

Repeat with new angles to get near

$$M_p = \max_{\vec{\gamma}, \vec{\beta}} F_p(\vec{\gamma}, \vec{\beta})$$

Example Max Cut on 3-regular Graphs



Assign ±1 to each vertex to maximize the number of edges on which the bit values disagree

Random Guessing gives 0.5

The QAOA (E.F., Goldstone Gutmann 2014) achieves 0.6924 for ALL instances!!!

$\boxed{P=1}$

not better than best classical

For p fixed we can classically preprocess and determine the best angles in advance.

Example: Max Cut on 3-regular graphs

$$C = \sum_{\langle jk \rangle} C_{\langle jk \rangle}$$

$$C_{\langle jk \rangle} = \frac{1}{2}\left(-Z_j Z_k + 1\right) \qquad Z_j = \pm 1$$

$p=1$ Look at contribution from edge $\langle jk \rangle$

$$\langle s| e^{i\gamma C} e^{i\beta B} Z_j Z_k e^{-i\beta B} e^{-i\gamma C} |s\rangle$$

$$|s\rangle = |+\rangle_1 |+\rangle_2 \cdots |+\rangle_n$$

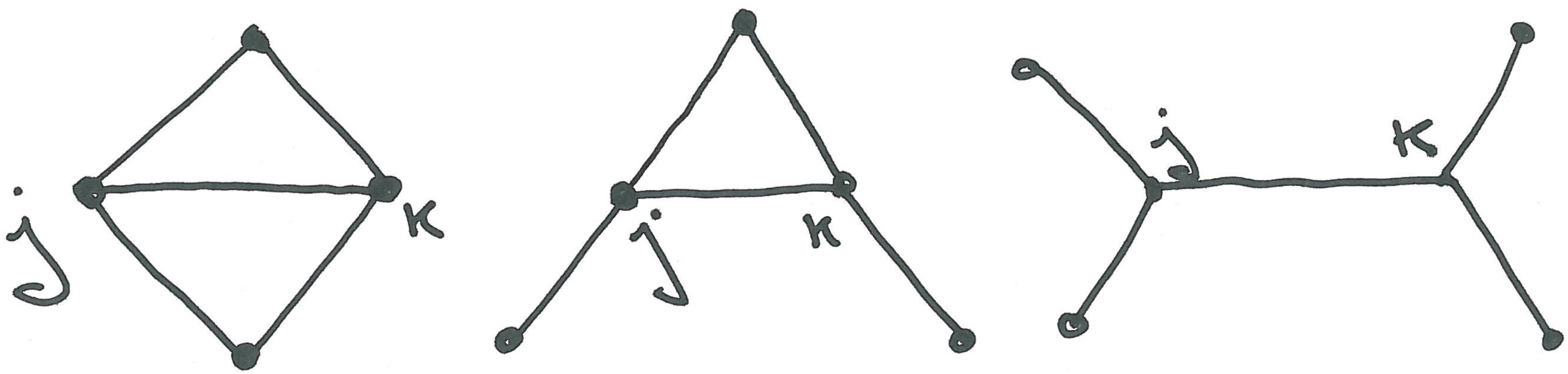
$$e^{i\beta B} Z_j Z_k e^{-i\beta B} = e^{i\beta(X_j + X_k)} Z_j Z_k e^{-i\beta(X_j+X_k)}$$

$$= [\cos 2\beta\, Z_j + \sin 2\beta\, Y_j][\cos 2\beta\, Z_k + \sin 2\beta\, Y_k]$$

only bits j k involved

Conjugate with $e^{i\gamma C}$

only bits connected to j, k involved



3 possible subgraphs

Each subgraph type gives a function of γ, β which does not depend on n or m.

These can be evaluated on a classical computer looking at a 4, 5 or 6 qubit system. Then

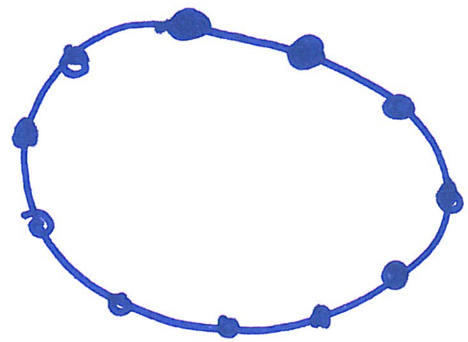
$$F_1(\gamma, \beta)$$

can be evaluated on a classical computer and optimal angles chosen.

At $p=1$, the QAOA will produce a cut that is at least .6924 times the optimal cut. For ALL instances of 3-regular Max Cut!

The .6924 improves on random guessing .5 but is not as good as the best classical algorithms.

Ring of Disagrees
(Max Cut)



n-bits

The QAOA gives an approximation ratio of $\frac{2p+1}{2p+2}$ independent of n

Note, at p=1 we get 3/4 an for large n almost every string satisfies 3/4 of the bonds.

Max E3LIN2

n variables, m equations each with 3 variables $(z_i + z_j + z_k) \bmod 2 = \begin{cases} 1 \\ 0 \end{cases}$ $\quad z_i = 0,1$

Instance is specified by a collection of triples and a 0 or 1 for each triple

Task Find a string that maximizes the number of satisfied equations. NP hard to find the optimal solution.

Try to find a "good" solution.

Approximate Optimization

Algorithm: Guess a Random String
Achieves $\frac{1}{2}$

Limit: $\left(\frac{1}{2}+\varepsilon\right)$ would imply P=NP 2001

Bounded Occurrence: Every variable is in no more than D equations

Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D}\right)$ 2000

Quantum Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D^{3/4}}\right)$ 2014

Classical Algorithm: $\left(\frac{1}{2}+\frac{\text{const}}{D^{1/2}}\right)$ 2015

Quantum Approximate Optimization Algorithm

E.F., Jeffrey Goldstone, Sam Gutmann

Better Analysis: $\left(\frac{1}{2} + \frac{1}{101\, D^{1/2} \ln D}\right)$

2015

Typical: $\left(\frac{1}{2} + \frac{1}{2\sqrt{3e}\, D^{1/2}}\right)$

Can we get rid of the $\ln D$???

If $\exists$ algorithm $\left(\frac{1}{2} + \frac{\text{const}}{D^{1/2}}\right)$

for a sufficiently large constant

then P=NP.

The QAOA exhibits Quantum Supremacy!!

with Aram Harrow

$$\tilde{H} = e^{-i\frac{\pi}{4}\sigma_x}$$

C = a sum of two bit clauses

$$|\psi\rangle = \tilde{H}^{\otimes n} e^{-i\gamma C} |s\rangle \qquad p=1$$

$$q(z) = |\langle z|\psi\rangle|^2$$

Suppose You have a classical algorithm that outputs strings with probability $p(z)$ and

$$|q(z) - p(z)| < q(z)/10$$

Then Polynomial Hierarchy Collapses!

Fault Tolerance

with
Adam Bookatz
Peter Shor

Simple Error Model

Between every desired gate there is a random unitary acting on each qubit

$$U_{error} = \prod_{j=1}^{n} U^{j}_{error}$$

$$U^{j}_{error} = e^{i \varepsilon_j \hat{n}_j \cdot \vec{\sigma}_j}$$

$\hat{n}_j$ chosen uniformly at random

ε_j small

Return to Max Cut $p=1$ example

$$|\gamma,\beta\rangle \longrightarrow U_{error}\, e^{-i\beta B}\, U_{error}\, e^{-i\gamma C}\, U_{error}\, |s\rangle$$

Now $\langle\gamma,\beta|\, Z_j Z_k \,|\gamma,\beta\rangle$ involves at most 6 qubits

$\langle\gamma,\beta|\, Z_j Z_k \,|\gamma,\beta\rangle$ gets reduced by

f^6 where $f = 1 - \frac{4}{3} E[\sin^2 \varepsilon]$

Approximation Ratio is reduced by an n independent factor !!

Fidelity goes down exponentially in n.

Much worse for big p and D but still no n!

More General Approach

Have different angles β for each qubit.

$\{\beta_i\}$ $i = 1$ to n for each p

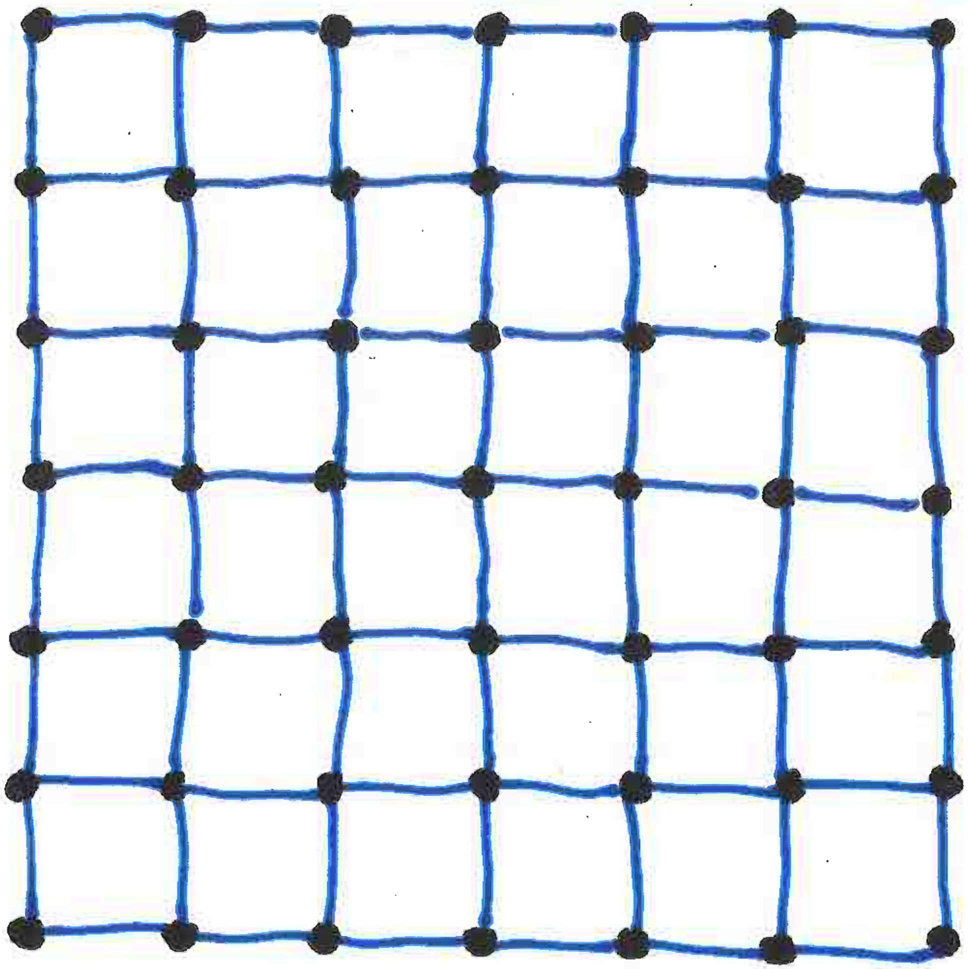
Have different angles γ for each clause

$\{\gamma_\alpha\}$ $\alpha = 1$ to M for each p

Or Use other Parameterized Unitaries

$$|\vec{\theta}\rangle = U_L(\theta_L) \cdots U_1(\theta_1)|+\rangle$$

Near Term Quantum Computer



 Single Qubit Unitary

 Two Qubit Unitary

Use the unitaries from the grid to make the state

$$|\vec{\theta}\rangle = U_L(\theta_L) \cdots U_1(\theta_1)\,|+\rangle$$

Depth limited by Device Coherence

Maximize $\langle \vec{\theta} | C | \vec{\theta} \rangle$

w/ Goldstone, Gutmann, Neven

Look at QAOA

Max Cut on 3-regular graphs

$$C = \sum_{\langle ij \rangle \text{ edge in graph}} \frac{1}{2}(1 - Z_i Z_j)$$

$$G = \sum_{\langle ij \rangle \text{ edge in Grid}} \frac{1}{2}(1 - Z_i Z_j)$$

$$U = e^{-i\beta B} e^{-i\gamma G}$$

$P = 1$

two angles γ, β

Provable Approximation ratio .53...

This approach solves the embedding problem!

U's do not depend on C.

After producing $|\vec{\theta}\rangle$ measure to get a string z and classically compute $C(z)$.

Pick new values of $\vec{\theta}$ and try to go uphill.

Will this work????

Run it and see!!!

Is this a Quantum Neural Net?

a Quantum Brain?