QAOA:

This is an optimization problem. Try to optimize an NP - Complete Problem.

PROBLEM: MaxSat [NP- Complete]

INTUITION: n variables, m clauses: Satisfy as many clauses as possible.

GIVEN: $\bigwedge_{j=1}^{m} (l_{j1} \vee l_{j2})$, int k (can we satisfy at least k clauses)

Boolean variables (or negation)

EXAMPLE: $(x_1 \vee x_2) \wedge (x_2 \vee \neg x_5) \wedge (\neg x_3 \vee \neg x_4) \wedge \cdots$

n BOOLEAN VARIABLES: $z \in \{0,1\}^n$

$$\text{count}_j(z) = \begin{cases} 1, & \text{if } z \text{ satisfies the } j^{TH} \text{ clause} \\ 0, & \text{otherwise} \end{cases}$$

does z (boolean string) Satisfy j^{TH} clause.

↳ combination of n boolean variables.

$$\text{count}(z) = \sum_{j=1}^{m} \text{count}_j(z).$$

$$C_j |z\rangle = \text{count}_j(z)\, |z\rangle$$

$$C\, |z\rangle = \text{count}(z)\, |z\rangle$$

$$= \sum_{j=1}^{m} \text{count}_j(z)\, |z\rangle$$

$$= \sum_{j=1}^{m} C_j\, |z\rangle$$

HERMITIAN MATRICES

$$B = \sum_{k=1}^{n-1} \text{NOT}_k$$

↳ X Gate on k^{TH} Qubit

SEPARATOR:

Mark the good cases. [Built from $C|z\rangle$]

MIXER:

Boost the amplitude of the marked cases. [Built from B]

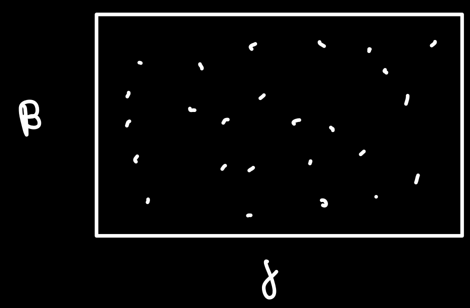
$$(\gamma, \beta) \in [0, 2\pi] \times [0, \pi]$$

$$\text{Sep}(\gamma) = e^{-i\gamma C}$$

As C is hermitian, sep is unitary.

$$\text{Mix}(\beta) = e^{-i\beta B}$$

Now, we try all possible values of γ, β



QAOA:

for many (γ, β) {

QUANTUM

MEASURE:

$$\left(Mix(\beta)\ Sep(\gamma)\ \underbrace{H^{\otimes n}}_{\text{all possible combinations}} |0^n\rangle \right)$$

Mark the good ones

Boost the marked good ones.

}

Pick a good one

EXAMPLE:

$$(x_0) \wedge (x_0 \wedge x_1)$$

WHAT IS $Sep(\gamma)$:

$$Sep(\gamma) = e^{-i\gamma C}$$

$$= e^{-i\gamma \sum c_j}$$

$$= \prod_j e^{-i\gamma c_j}$$

$\rightarrow$ c_j's commute

$$e^{-i\gamma c_j}|z\rangle = \begin{cases} e^{-i\gamma}|z\rangle, & \text{if } count_j(z)=1 \\ |z\rangle, & \text{if } count_j(z)=0 \end{cases}$$

mark

$$Sep(\gamma) \sum_{z\in\{0,1\}^n} a_z|z\rangle$$

$\longrightarrow$ amplitudes

$$= \prod_j e^{-i\gamma c_j} \sum_z a_z|z\rangle$$

$$= \sum_z a_z \underbrace{\prod_j e^{-i\gamma c_j}}|z\rangle$$

if $count_j(z)=1$, it remains so it is multiplied count(z) times.

$$= \sum_z a_z \boxed{(e^{-i\gamma})^{count(z)}}|z\rangle$$

EXAMPLE:

$(x_0) \wedge (x_0 \wedge x_1)$

$$\mathrm{sep}(\gamma)\,(a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle)$$

$$= a_{00}|00\rangle + a_{01}|01\rangle + \underbrace{a_{10}(e^{-i\gamma})^1|10\rangle + a_{11}(e^{-i\gamma})^2|11\rangle}_{\text{GOOD CASES MARKED}}.$$

$\gamma = \beta = \pi/4$:

00 : 3.7%

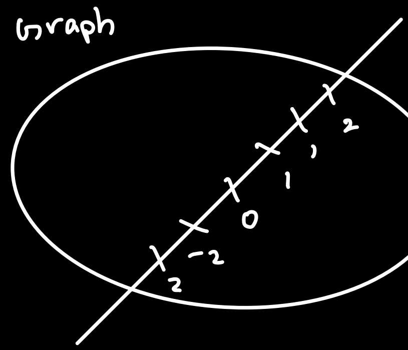
01 : 3.7%

10 : 28.7%

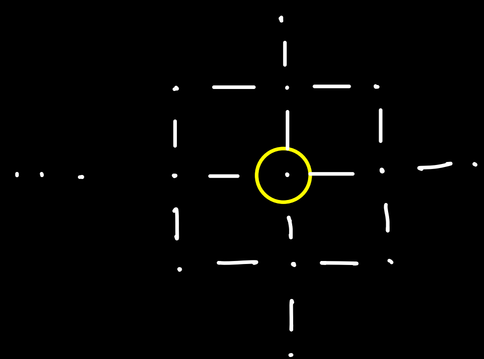
11 : 64.0%

GOOGLE'S PAPER:

MAX CUT:



Sycamore Quantum Computer:



Graphs: (Types of graphs they work with in the paper)

1. Subgraphs of the hardware graph] Most used

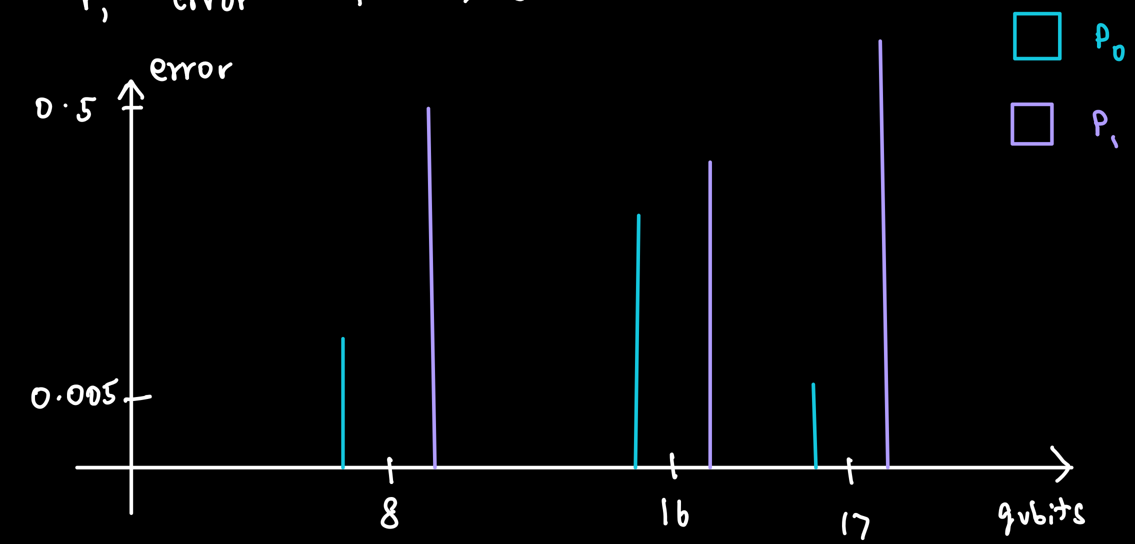
2. 3 - regular graph
   (connected to 3 other nodes)] known NP complete with least edges

3. Fully connected graph.] Most edges case

BIAS IN MEASUREMENT :

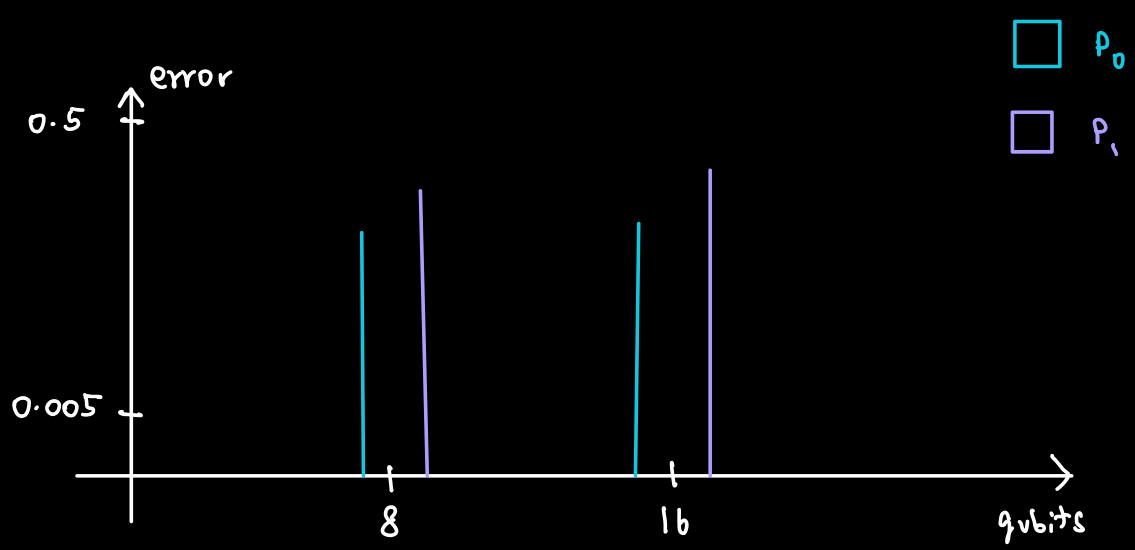
P_0 error : $\underset{\text{expected}}{0} \longrightarrow \underset{\text{observed}}{1}$

P_1 error : $1 \longrightarrow 0$



So the error is random for different qubits. No idea which are nice.

So we run multiple times and post process

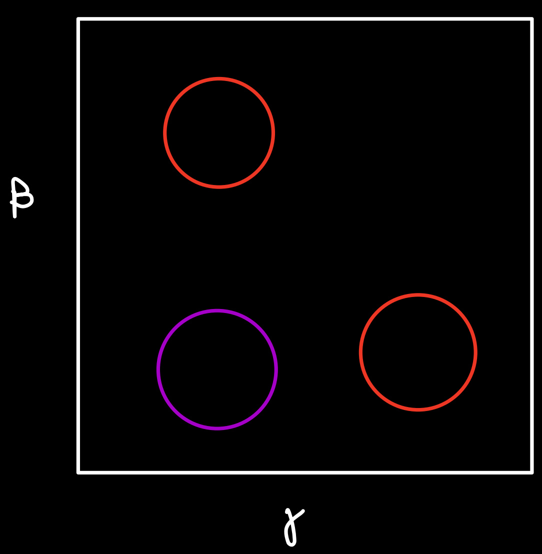


This helps them get closer.

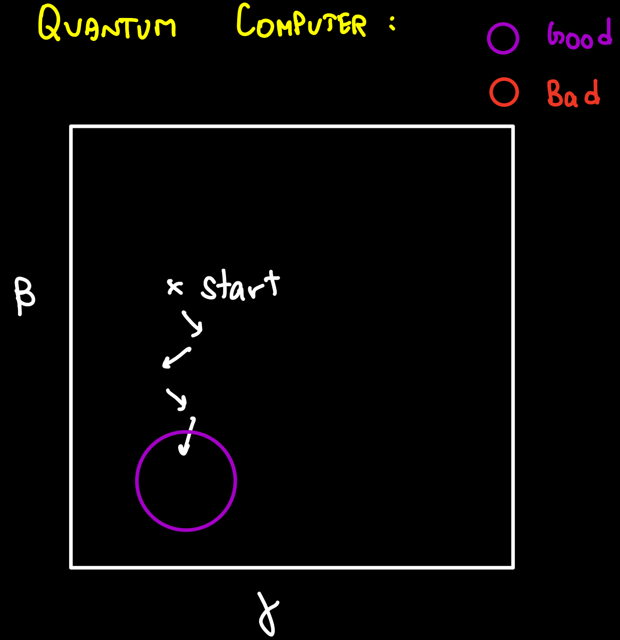
0 → 1 and 1 → 0 are equally likely and reduces the bias.

note: P_1 by nature is mostly larger than P_0 so we observe more 0s than expected.

SIMULATION :



QUANTUM COMPUTER :

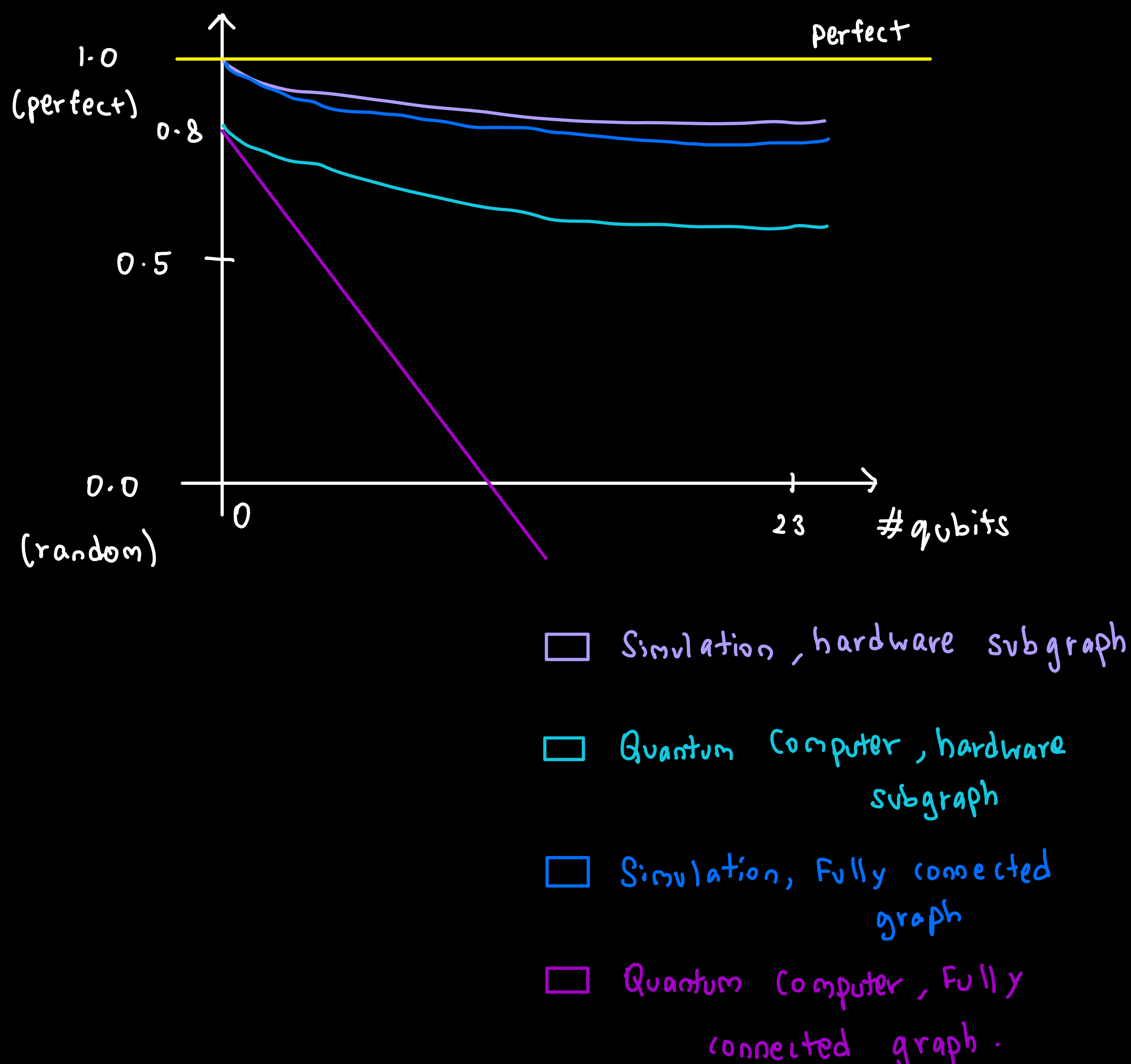


→ 50,000 times done for each γ, β ← repeat

(5000 per second is the speed)

→ Then do a kind of gradient descent

→ Move to new γ, β



So though theoretically and in simulation, repeatedly doing separator, mixer improves results.

But in practice error outweighs the optimization.