FR1.R6.4: Classical Meets Quantum

Channel Decoding with Quantum Approximate Optimization Algorithm

Toshiki Matsumine

Toshiaki Koike-Akino

Ye Wang

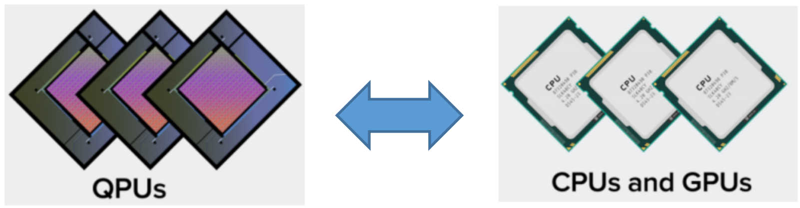
July 12, 2019

MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com

MITSUBISHI ELECTRIC
Changes for the Better

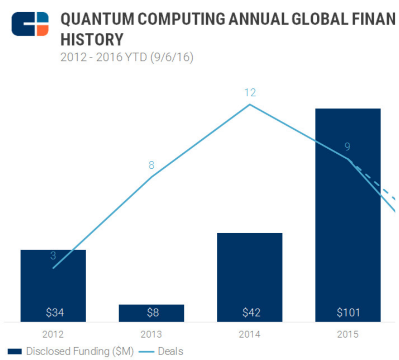
Agenda

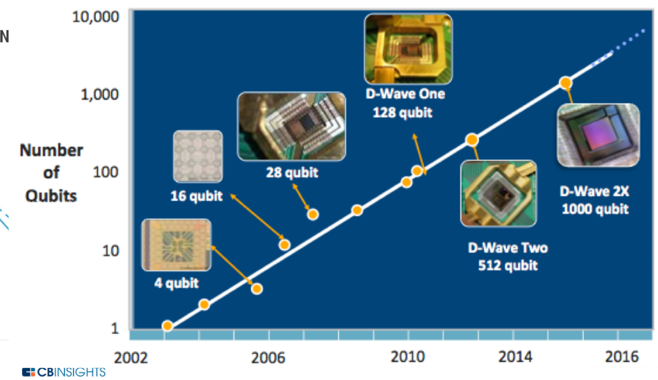
- Background of quantum technology trend
- Variational quantum algorithms
- Quantum Approximate Optimization Algorithm (QAOA)
- QAOA hybrid quantum-classical channel decoding
- Simulation and real quantum processor results
- New theory
- Degree optimization for QAOA-friendly channel codes
- Summary



Background

- Morgan Stanley: Quantum tech can drive **4th industrial revolution**
- Escalating government funds: National Quantum Initiative **$1.2B**
- Quantum providers: **IBM**, **Google**, **Microsoft**, **Honeywell**, Intel, Nokia, AirBus, IONQ, rigetti



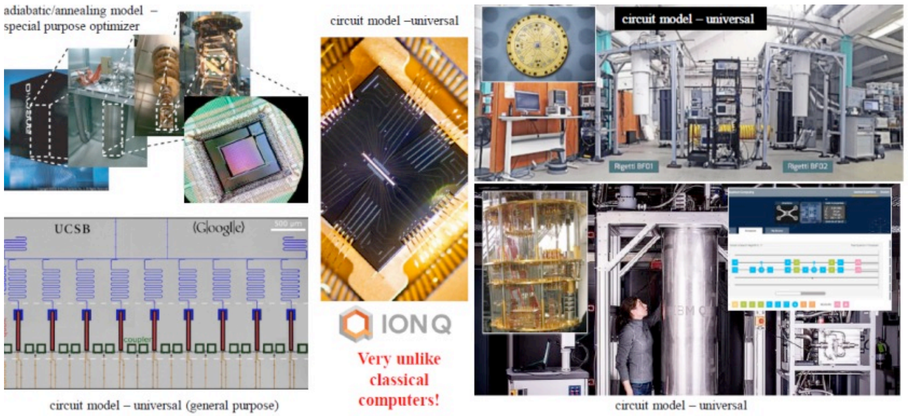


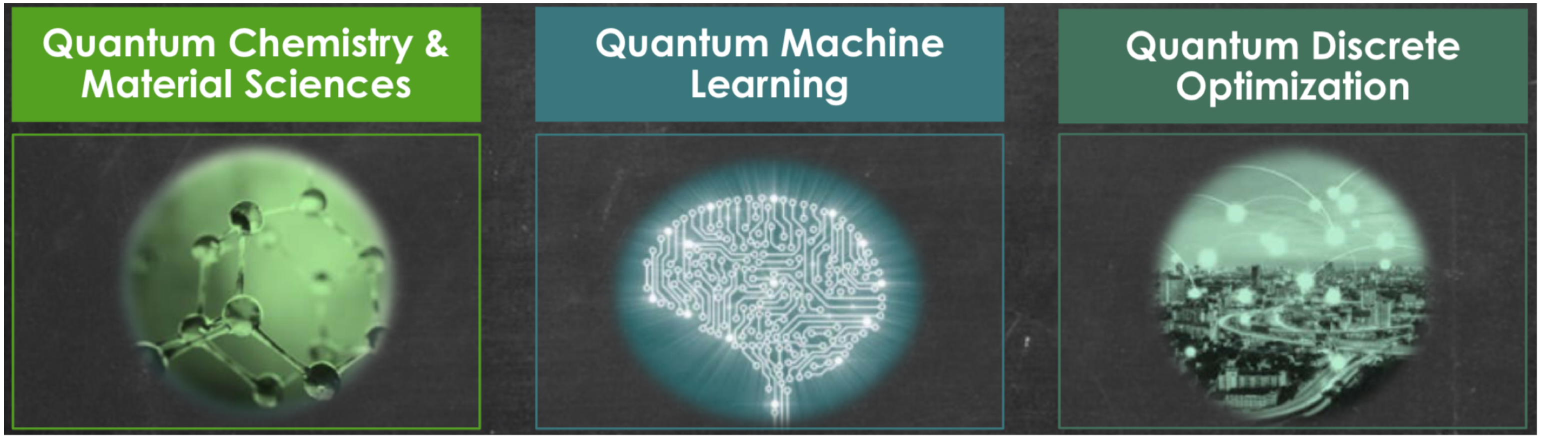
Free Python libraries to try quantum computing on realistic simulators or real devices

Quantum Computing … Not Far Future

- Quantum processing units (QPU) are already in front of us

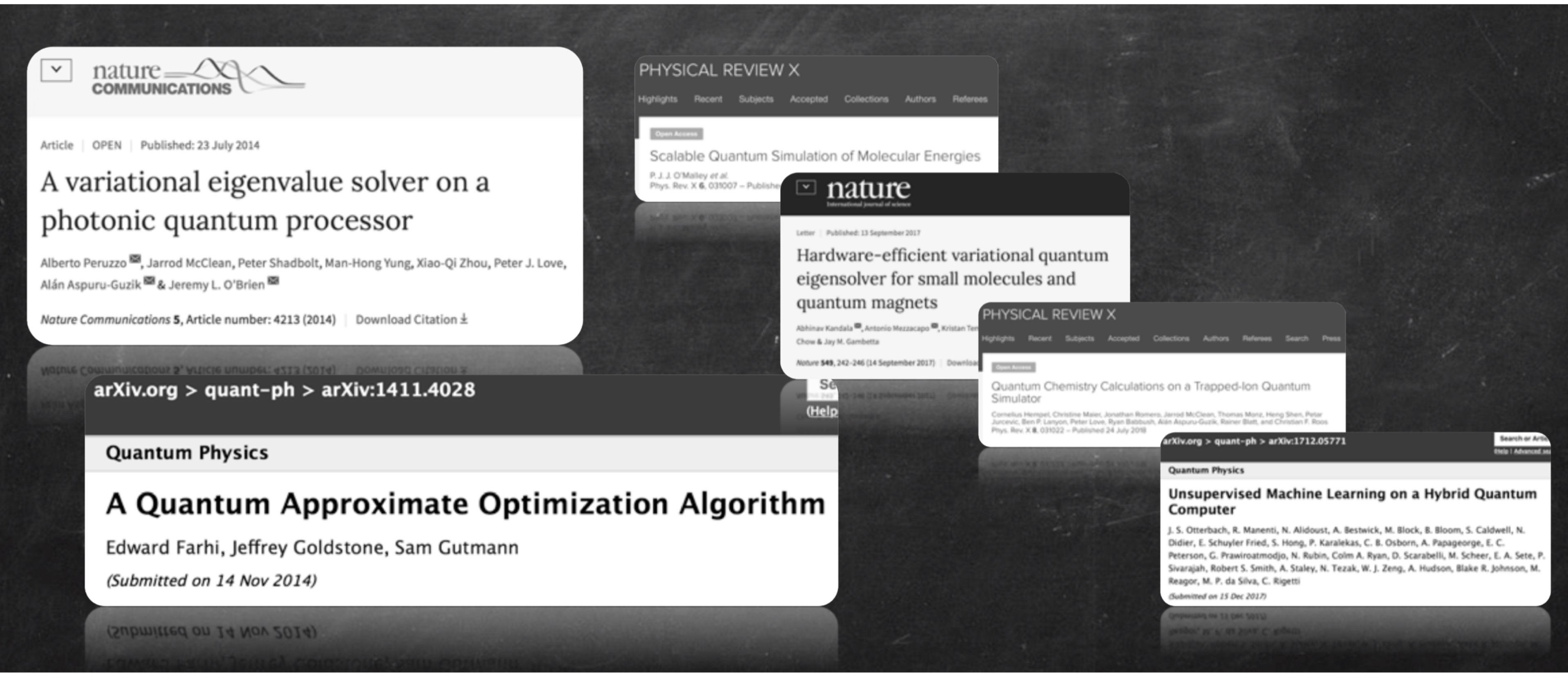
- Various applications
  - Chemistry, material science
  - Machine learning
  - Optimization





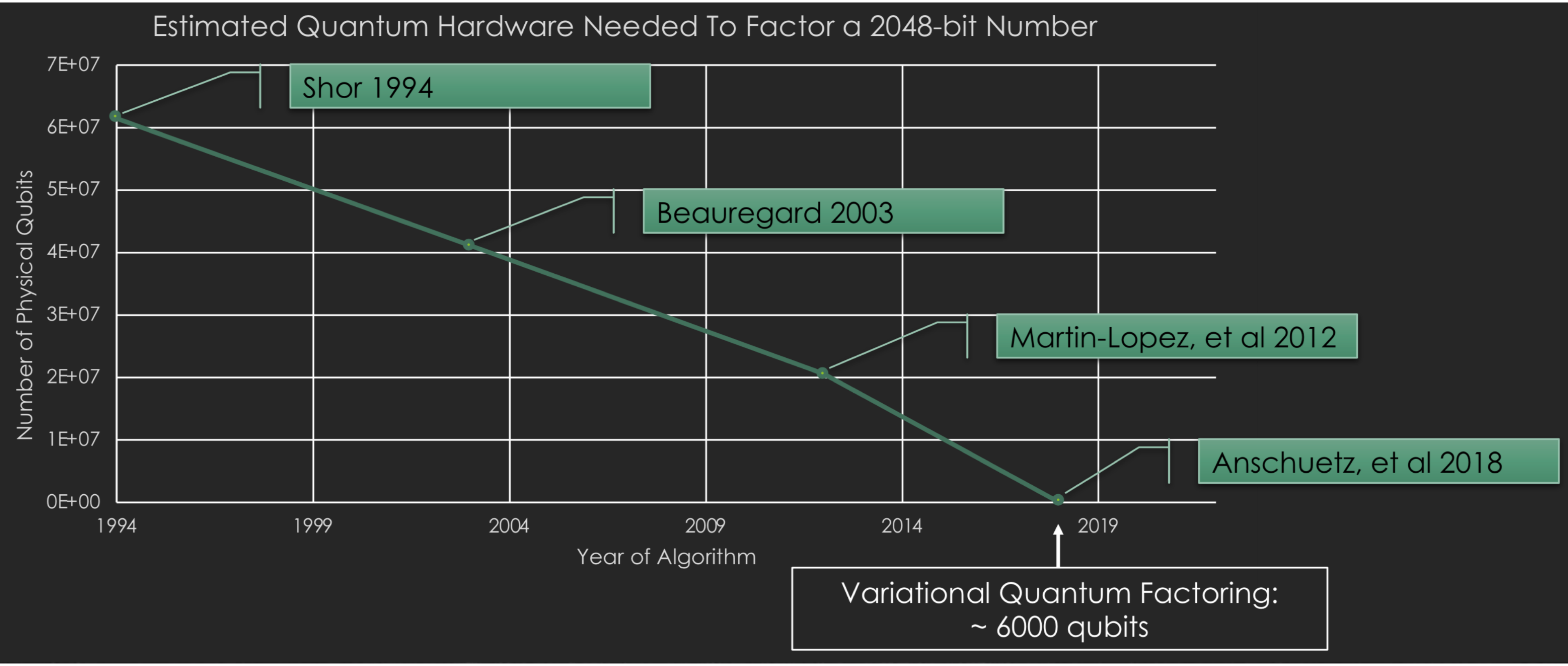
Post-2014 Trend: Variational Quantum Principle

- Hybrid use of quantum measurement and classical optimization
  - VQE: Variational Quantum Eigensolver (2014)
  - QAOA: Quantum Approximate Optimization Algorithm (2014)



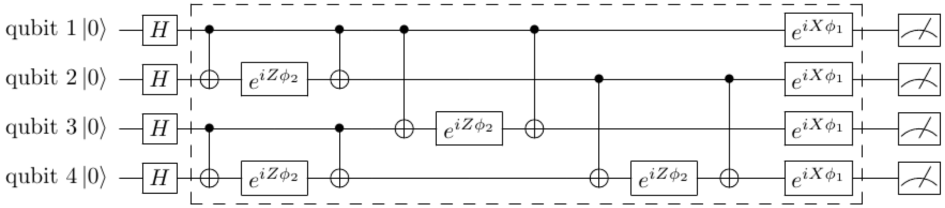
Example: Variational Quantum Factoring (VQF)

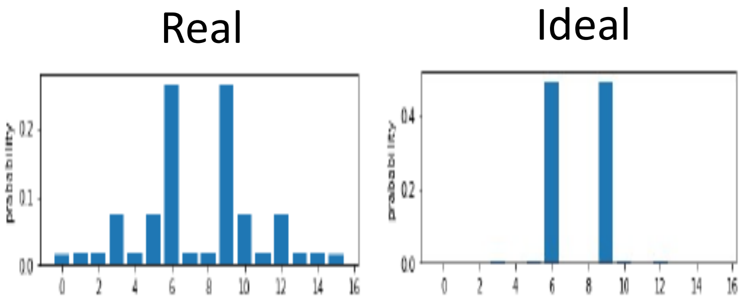
- Quantum factoring by Shor's algorithm (1994) showed super-polynomial speed-up, … but requires noise-free quantum gates
- VQF (2018) can reduce required qubits by 4 orders of magnitude, by removing necessity of error corrections



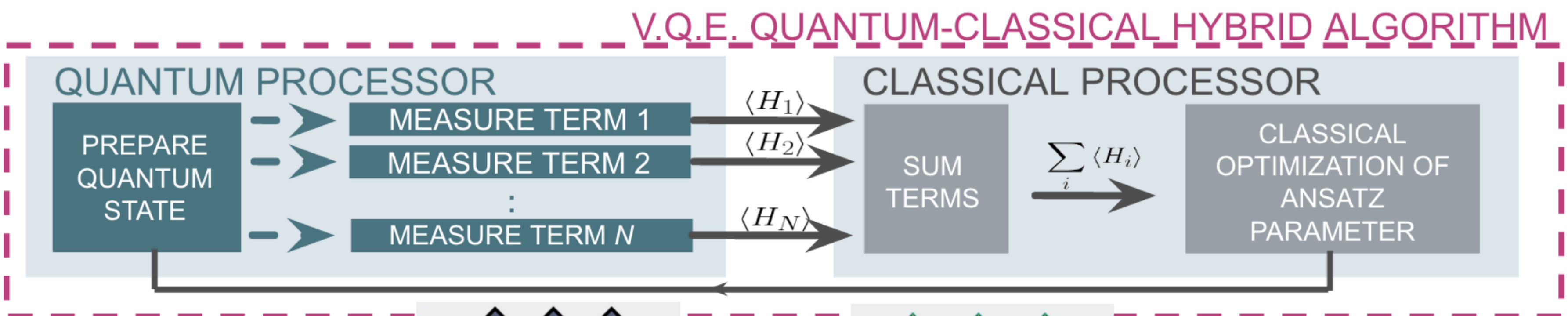
Variational Quantum Algorithms for NISQ

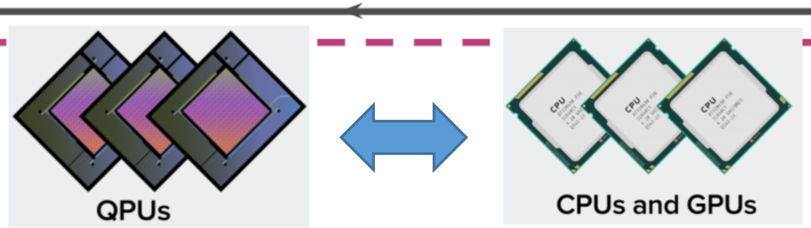
- Current quantum processors are noisy and limited-coherent: quantum gates cannot be perfect





- For noisy intermediate-scale quantum (**NISQ**) devices, variational hybrid quantum-classical algorithms may be a viable driver for quantum supremacy due to shallow gates and noise resilience



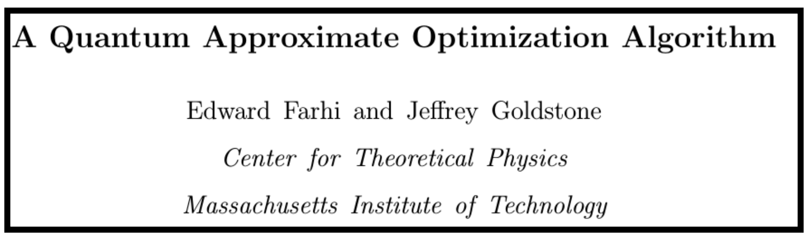


QAOA: Quantum Approximate Optimization Alg.

- Alternating cost Hamiltonian and mixer Hamiltonian like annealing
- Convergence theorem to eigenstate for infinite-level QAOA
  - Infinite Suzuki-Trotter decomposition with adiabatic annealing
- Classical optimization of variational angle parameters given quantum measurement

$$\lim_{p\to\infty} F_p^\star = \max_{\mathbf{z}} C(\mathbf{z})$$

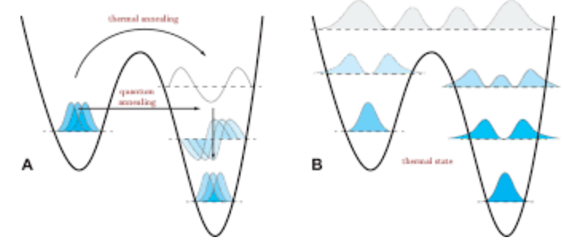
- Theoretical analysis showed better accuracy than classical counterparts; e.g. MaxCut, MaxSat, MaxClique

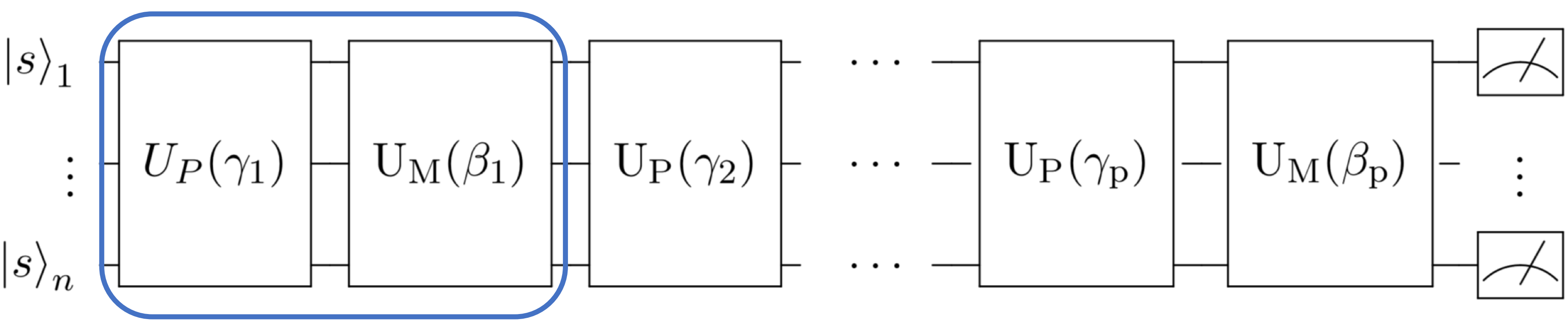

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone

Center for Theoretical Physics

Massachusetts Institute of Technology

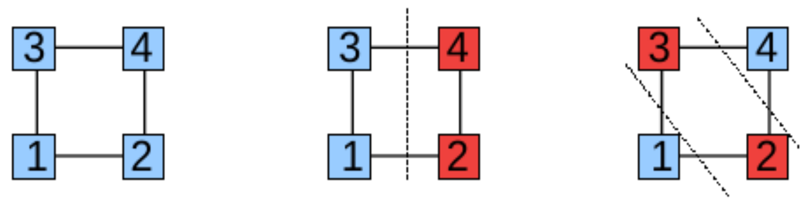




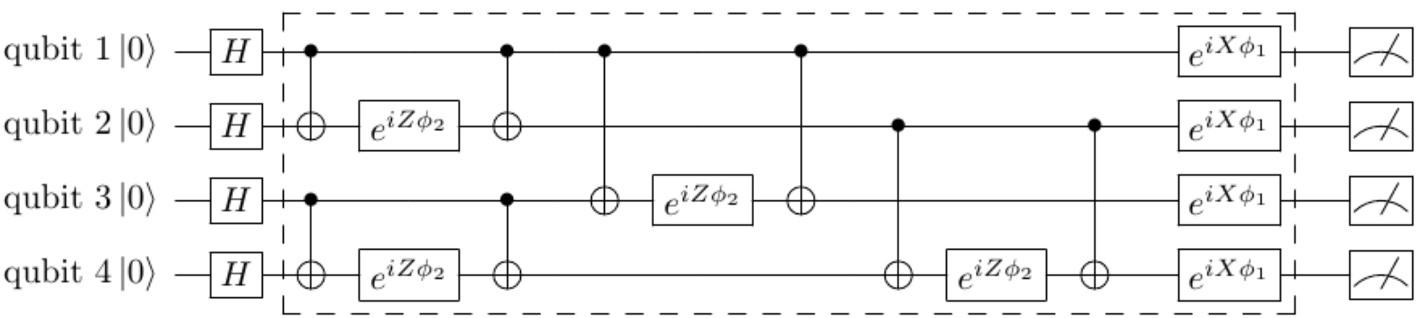
Example: QAOA for MaxCut Problem

- Construct Hamiltonian to maximize the cut

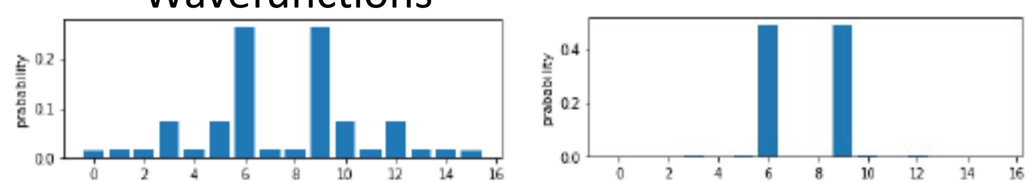
$$C = \sum_{(uv)\in E} C_{uv}, \qquad C_{uv} = \frac{1}{2}(I - Z_u Z_v)$$



[1111]: weight 0 [1010]: weight 2 [1001]: weight 4



Wavefunctions



QAOA-1 MaxCut Theory

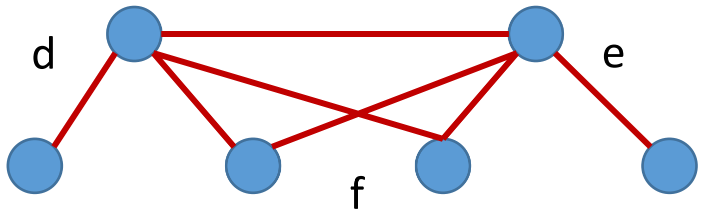
- Theorem (2018): For level-1 QAOA, approximation performance depends on graph edge degrees:

- *For each edge* (uv),

$$\begin{aligned}\langle\gamma\beta|C_{uv}|\gamma\beta\rangle &= \frac{1}{2}+\frac{1}{4}\sin(4\beta)\sin\gamma\,(\cos^d\gamma + cos^e\gamma)\\ &- \frac{1}{4}\sin^2\beta\cos^{d+e-2f}\gamma\,(1-\cos^f 2\gamma)\end{aligned}$$

where $d = deg(u)-1$, $e = deg(v)-1$, *and* f *is the number of triangles in the graph containing* (uv).

$$\max_{\gamma,\beta}\langle C\rangle \geq \frac{m}{2}+\frac{m}{2\sqrt{e}}\frac{1}{\sqrt{D_G}} - O\left(\frac{F}{D_G}\right)$$

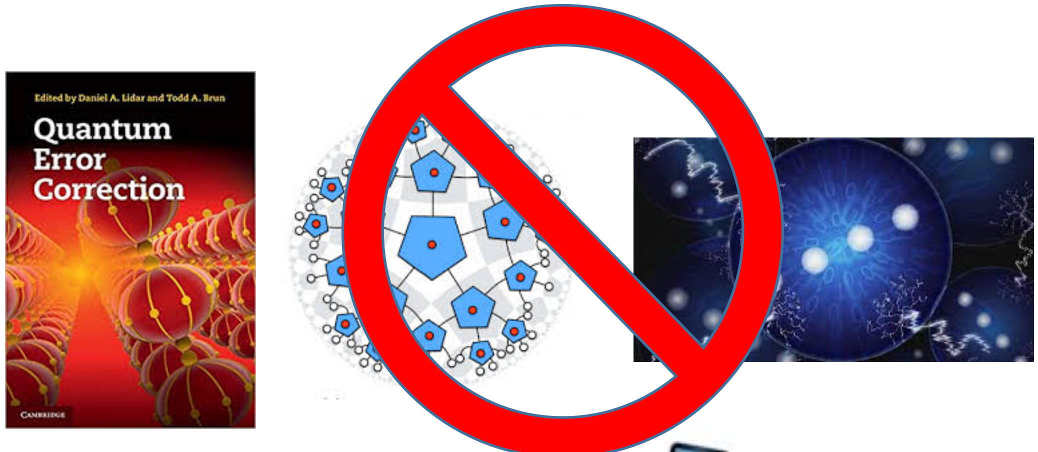


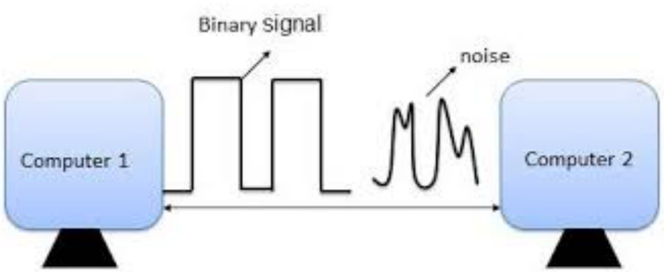
Expected solution outperformed
best-known classical algorithms, Goemans-Williamson (1995)

Triangle =
Girth-6 in factor graph

Application to Channel Decoding Problem

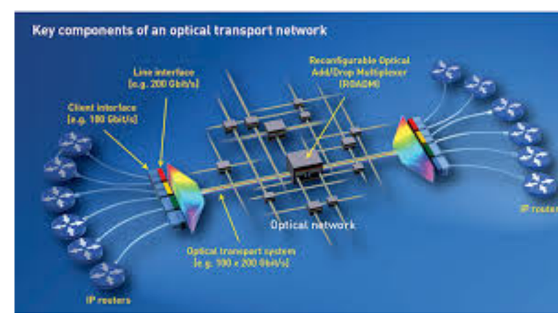
- This talk is **not** “Quantum Error Correction Codes (**QECC**)” to correct quantum errors in quantum channels/systems
- We want to correct classical errors with classical error correction codes (ECC) in classical channels through the use of CPU and QPU





Wired Network

Radio Network



Fiber-Optic Network

Classical Channel Decoding

- Hamming codes, Reed-Muller codes, Golay codes, convolutional codes, turbo codes, low-density parity-check (LDPC) codes, polar codes, …
- Suppose linear binary codes with generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$

Redundancy: Parity

[01011100101011] → [01011100101011 00100101110111011]

$\mathbf{u} \in \mathbb{F}_2^k$ $\quad$ $\boxed{\mathbf{x} = \mathbf{u}\mathbf{G}}$ $\quad$ $\mathbf{x} \in \mathbb{F}_2^n$

- Communication channel exhibits noise

$$\mathbf{y} = \mathbf{x} + \mathbf{w} \qquad \mathbf{w} \in \mathbb{R}^n$$

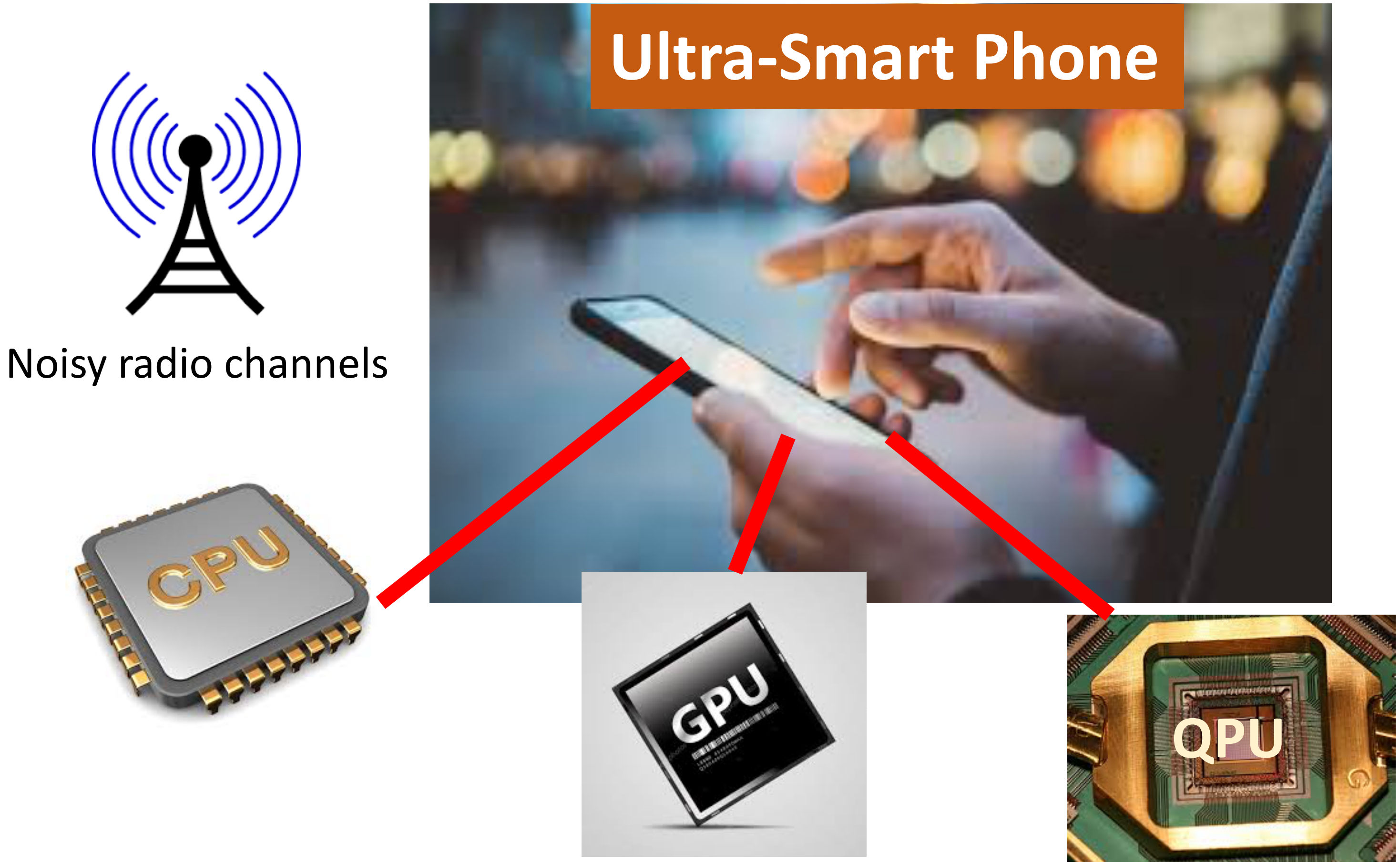
- Maximum-likelihood (ML) decoding for symmetric channels:

$$\arg\min_{\mathbf{u}} d_{\mathrm{H}}(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{u}} \sum_{\nu=1}^{n} (1 - 2y_\nu)(1 - 2x_\nu)$$

NP-hard 2^k search for maximum correlation

Hybrid Quantum-Classical Channel Decoding

- Suppose we have ultra-smart phone, equipped with CPU, GPU/TPU, and QPU
- We use VQE/QAOA to realize quasi-ML decoding for reliable telecommunications



QAOA Channel Decoding

- Convert ML decoding problem into Ising Hamiltonian model

$$\arg\min_{\mathbf{u}} d_{\mathrm{H}}(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{u}} \sum_{\nu=1}^{n} (1-2y_\nu)(1-2x_\nu)$$

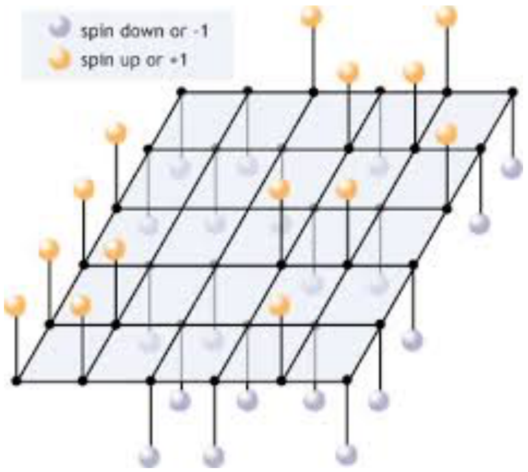
$$\mathbf{x} = \mathbf{u}\mathbf{G}$$

k-bit search: 2^k

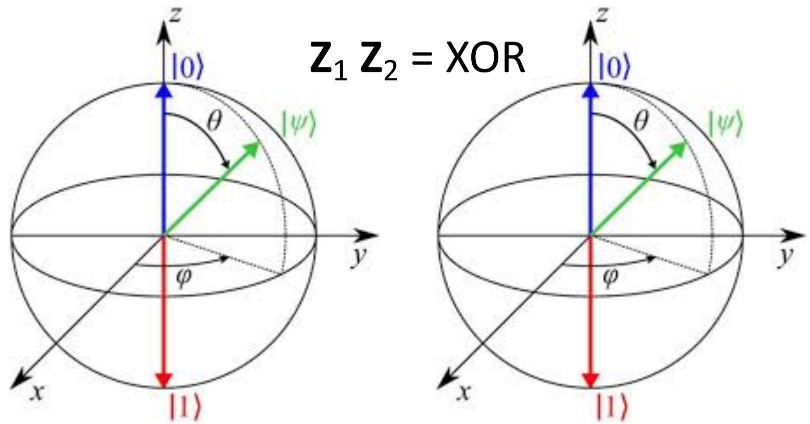
k-qubit parallel operation

$$C = \sum_{\nu=1}^{n} C_\nu = \sum_{\nu=1}^{n} (1-2y_\nu) \prod_{\kappa \in \mathcal{I}_\nu^{\mathrm{c}}} \mathbf{Z}_\kappa$$

Pauli-Z



$\mathbf{Z}_1\,\mathbf{Z}_2$ = XOR

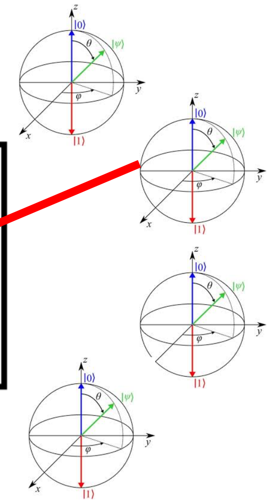


Example: Hamming Code Hamiltonian

- [7, 4] Hamming code is best-known code for n=7 and k=4, having minimum Hamming distance of 3, which can correct 1 bit error
- Generator matrix:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Degree: $[1, 1, 1, 1, 3, 3, 3]$



$$C = r_1\mathbf{Z}_1 + r_2\mathbf{Z}_2 + r_3\mathbf{Z}_3 + r_4\mathbf{Z}_4$$

Degree-1

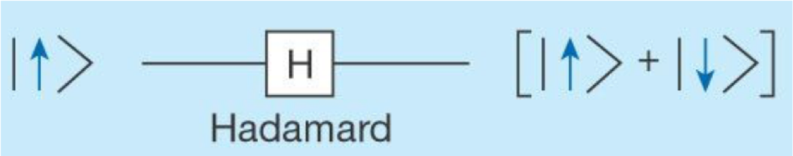
$$+ r_5\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4 + r_6\mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_4 + r_7\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$$

Degree-3

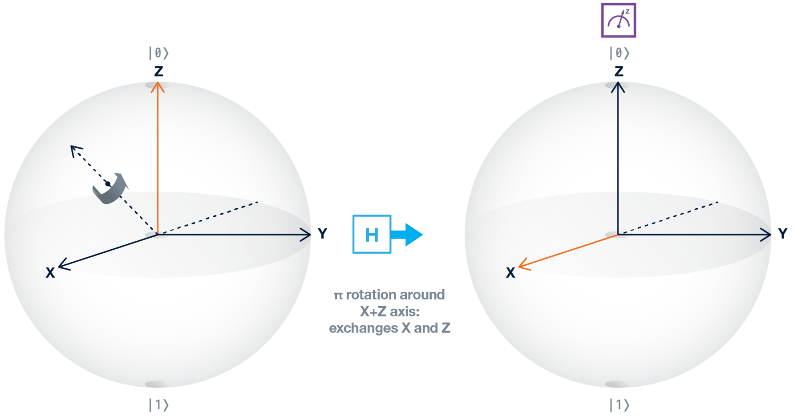
c.f.) MaxCut Hamiltonian is regular degree-2

Initial Quantum State and Mixer Hamiltonian

- We consider Hadamard superposition state
  - 50% chance of 0 or 1 measurement, thus **random** search



$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$



- We use admixing Hamiltonian for annealing
  - Why? Its eigenstate is Hadamard state

$$B = \sum_{\kappa=1}^{k} \mathbf{X}_{\kappa}$$

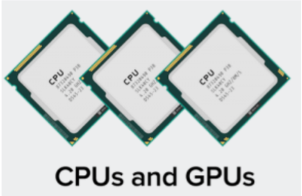
Pauli-X

Eigenstate:

$$|\phi\rangle = |+\rangle^{\otimes k}$$

QAOA Channel Decoding

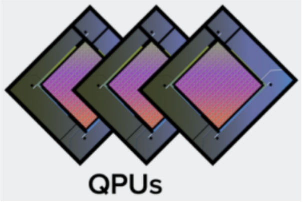
- CPU:
  - Given generator matrix G and received signal y
  - Construct cost Hamiltonian with variational angles
  - Quantum shots on QPU to obtain quasi-ML decision
  - Re-optimize angles if necessary and re-shot



$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$C = r_1\mathbf{Z}_1 + r_2\mathbf{Z}_2 + r_3\mathbf{Z}_3 + r_4\mathbf{Z}_4 + r_5\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4 + r_6\mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_4 + r_7\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$$

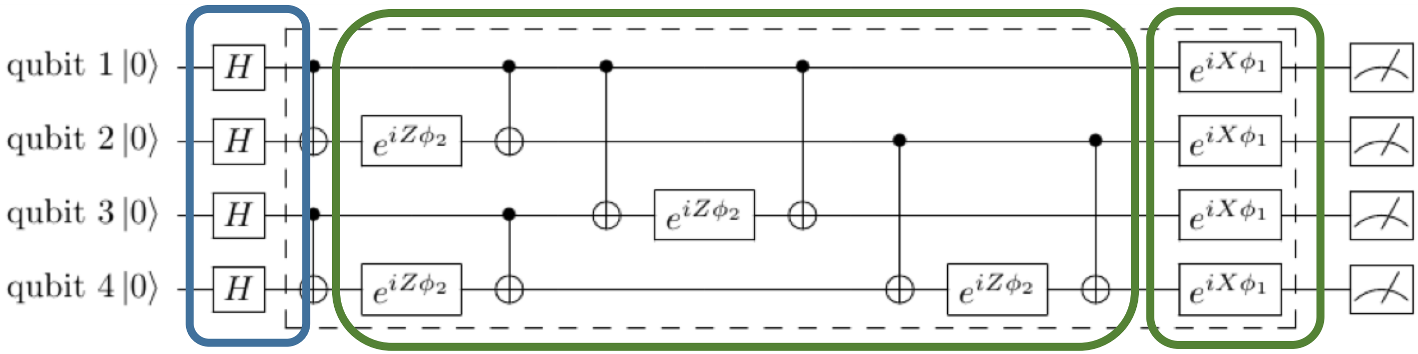
- QPU: QAOA
  - Initialize quantum state: |+> with Hadamard gates
  - Apply gamma angle rotation with cost Hamiltonian **C**
  - Apply beta angle rotation with mixer Hamiltonian **B**
  - Repeat *p*-times for level-*p* QAOA
  - Measure



$|s\rangle_1 \;\vdots\; |s\rangle_n$ — $U_P(\gamma_1)$ — $\mathrm{U_M}(\beta_1)$ — $\mathrm{U_P}(\gamma_2)$ — $\cdots$ — $\mathrm{U_P}(\gamma_\mathrm{p})$ — $\mathrm{U_M}(\beta_\mathrm{p})$ — measure

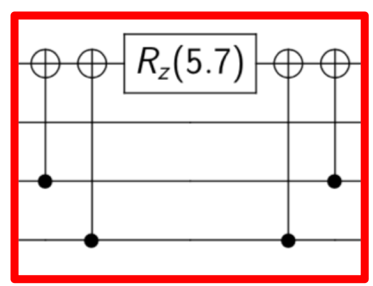
Quantum Circuits for QAOA Decoding

- State preparation: Hadamard *H*
- Mixer Hamiltonian operation: $\exp(\jmath\beta\mathbf{B})$
- Cost Hamiltonian operation: $\exp(\jmath\gamma\mathbf{C})$

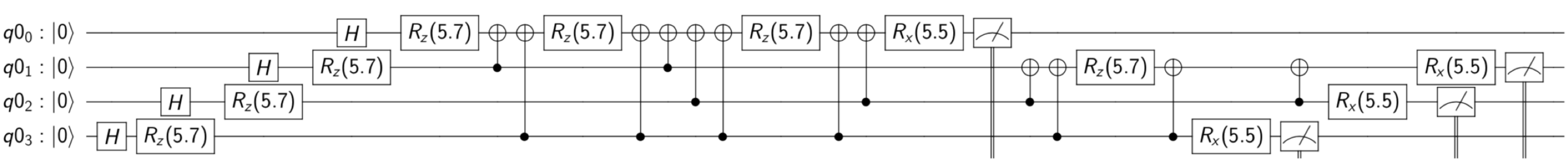


x *p*

Degree-*d* XOR: 2(*d*-1) CNOT

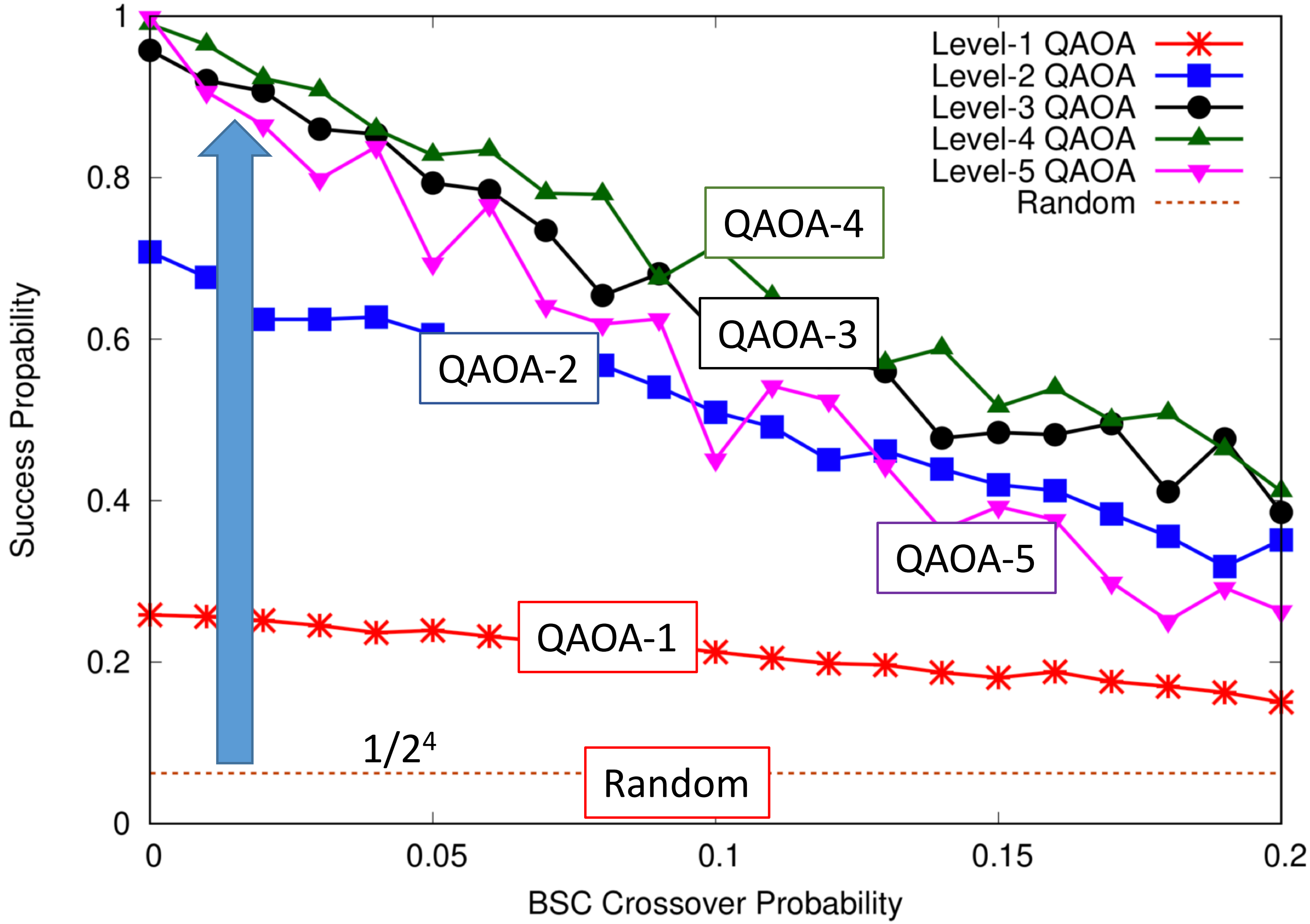


Hamming code QAOA decoder



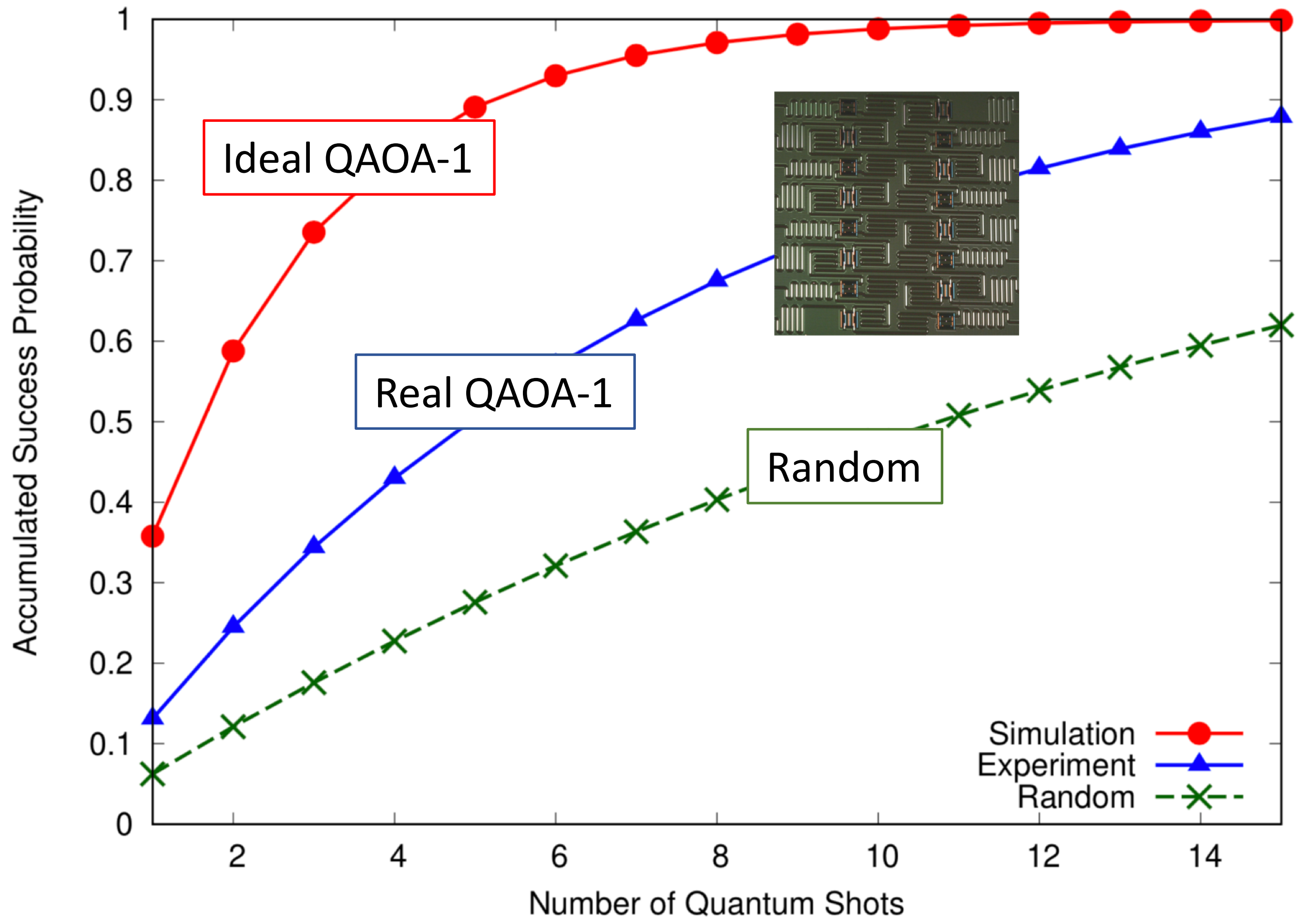
Quantum Simulation: Binary Symmetric Channel

- Rigetti pyquil is used for VQE with Nelder-Mead to optimize variational angles
- IBM qiskit is used for QAOA decoding simulation for validation



Multi-Shot Decoding: IBM Q14 Melbourne Chip

- ML decision success probability can be improved by taking multiple measurements of QPU shots



Theoretical Analysis of QAOA Decoding

- Given code generator matrix, derive cost expectation

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

- For QAOA ansatz states with variational angles beta and gamma

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

$$U(B, \beta) = \exp(-\jmath \beta B) \qquad U(C, \gamma) = \exp(-\jmath \gamma C)$$

- Cost Hamiltonian $$C = \sum_{\nu=1}^{n} C_\nu = \sum_{\nu=1}^{n} (1 - 2y_\nu) \prod_{\kappa \in \mathcal{I}_\nu^{\mathrm{c}}} \mathbf{Z}_\kappa$$

- Mixer Hamiltonian $$B = \sum_{\kappa=1}^{k} \mathbf{X}_\kappa$$

- Focus on each clause of cost Hamiltonian

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \sum_\nu \langle C_\nu \rangle$$

New Theorem

- For QAOA-1 decoding, cost expectation is expressed as follows:

$$F_1(\gamma_1, \beta_1) = \sum_{\nu=1}^{n} (1 - 2y_\nu) \sum_{\mathbf{b} \in \mathbb{F}_2^k} \sum_{\mathbf{a} \in \mathbb{F}_2^\rho : \mathbf{b} = \mathbf{G}^\mathbf{b} \mathbf{a}} A_\nu^{\mathbf{a},\mathbf{b}} (\jmath s)^\omega c^{(\rho-\omega)} (-\jmath s')^\varpi c'^{(d_\nu^c - \varpi)}$$

$$c' = \cos(2\beta_1) \text{ and } s' = \sin(2\beta_1)$$

$$c = \cos(2(-1)^y \gamma_1) \text{ and } s = \sin(2(-1)^y \gamma_1)$$

where rho is rank, omega is weight of vector **a**, pi is weight of vector **b**, $\mathbf{G}^\mathbf{b}$ is sub-generator matrix associated with **b**, and *A* is the number of conditional pairs subject to $\mathbf{b}=\mathbf{G}^\mathbf{b}\mathbf{a}$

- This theorem holds for any arbitrary linear binary codes

Mixer Hamiltonian (RX) on Cost Function

- Focus each clause in cost expectation $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \sum_\nu \langle C_\nu \rangle$

$$\langle C_\nu \rangle = \langle \phi | U^\dagger(C, \gamma_1) U^\dagger(B, \beta_1) C_\nu U(B, \beta_1) U(C, \gamma_1) | \phi \rangle$$

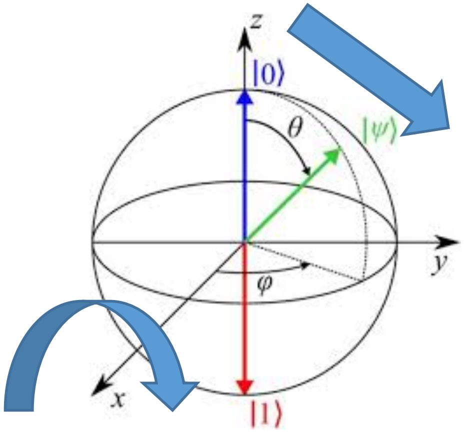
$$U(B, \beta_1)^\dagger \left(\prod \mathbf{Z}_\kappa\right) U(B, \beta_1) = \prod (c' \mathbf{Z}_\kappa + s' \mathbf{Y}_\kappa)$$

$$\prod \exp(-\jmath \beta_1 \mathbf{X}_i)$$

Recall Pauli rules:

$$\exp(\jmath\beta\mathbf{W}) = \cos(\beta)\mathbf{I} + \jmath\sin(\beta)\mathbf{W}$$
$$\mathbf{WW} = \mathbf{I}$$
$$\mathbf{XY} = \jmath\mathbf{Z}$$
$$\mathbf{YZ} = \jmath\mathbf{X}$$
$$\mathbf{ZX} = \jmath\mathbf{Y}$$

Cost Z



Mixer X-rotations

Binomial Expansion

- *d*-ary Product of binary Pauli sum:

$$U(B,\beta_1)^\dagger \left(\prod \mathbf{Z}_\kappa\right) U(B,\beta_1) = \prod^{d} (c'\mathbf{Z}_\kappa + s'\mathbf{Y}_\kappa)$$

- Expand to 2^d-ary sum of *d*-ary Pauli product:

$$c'^d \mathbf{Z}_1\mathbf{Z}_2\cdots\mathbf{Z}_d + c'^{d-1}s'\mathbf{Z}_1\mathbf{Z}_2\cdots\mathbf{Y}_d + \cdots s'^d\mathbf{Y}_1\mathbf{Y}_2\cdots c'\mathbf{Y}_d$$

- Let **b**={0,1}^{d} indicate expansion term entangled either *c'***Z** or *s'***Y**

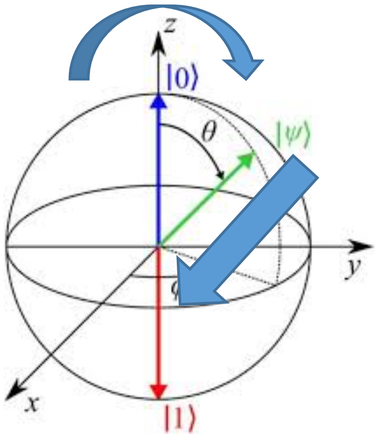
b=[0 0 0 … 0] b=[0 0 0 … 1] b=[1 1 1 … 1]

- Cost expectation will be proportional to

$$(-\jmath s')^{\varpi} c'^{(d^c_\nu - \varpi)} \text{ due to } \mathbf{ZY} = -\jmath\mathbf{X} \text{ and } \langle +|\mathbf{X}|+\rangle = 1$$

$$\varpi = |\mathbf{b}|_0$$

Cost Z-rotations



Admixing |+> measurement

Cost Hamiltonian (RZ) on Expanded Pauli Product

- Let $\mathbf{W}^{\mathbf{b}} = c'^{d-\varpi} s'^{\varpi} \mathbf{Z}_i \ldots \mathbf{Y}_j$ be expanded Pauli product associated with binary indicator **b**
- Conjugate with cost Hamiltonian can be expressed by non-commutable Hamiltonian $\mathbf{C}^{\mathbf{b}}$ of rank rho

$$U(C,\gamma_1)^{\dagger}\mathbf{W}^{\mathbf{b}}U(C,\gamma_1) = U(C^{\mathbf{b}}, 2\gamma_1)^{\dagger}\mathbf{W}^{\mathbf{b}}$$

- Non-commutable sub-generator matrix $\mathbf{G}^{\mathbf{b}}$ is column-selective **G** whose weight is odd after Hadamard product

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(1-2y_5)\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4$$

$$\mathbf{b} = [1,1,0,1]$$

$$\mathbf{W}^{\mathbf{b}} = s'^3\mathbf{Y}_1\mathbf{Y}_2\mathbf{Y}_4$$

$$\mathbf{bG} = [1,1,0,1,1,0,0]$$

$$\mathbf{G}^{\mathbf{b}} = [\mathbf{G}]_{:,\{1,2,4,5\}}$$

Binomial Expansion of Cost Hamiltonian

- Cost Hamiltonian conjugate on cost function:

$$U(C,\gamma_1)^\dagger \mathbf{W}^{\mathbf{b}} U(C,\gamma_1) = U(C^{\mathbf{b}}, 2\gamma_1)^\dagger \mathbf{W}^{\mathbf{b}}$$

- The non-commutable cost Hamiltonian operator is expressed

$$U(C^{\mathbf{b}}, 2\gamma_1)^\dagger = \prod_{\nu}^{\rho} \mathrm{e}^{2\jmath\gamma_1 C_\nu} = \prod_{\nu}^{\rho} (c\mathbf{I} + \jmath s \prod_{\kappa} \mathbf{Z}_\kappa)$$

- Again, take binomial expansion from rho-ary product to 2^{rho}-ary sum
- Let **a**={0,1}^{n} indicate binary selection of eather **cI** or js**Z**$_1$ **Z**$_2$...

$$\mathbf{G}^{\mathbf{b}} = [\mathbf{G}]_{:,\{1,2,4,5\}} \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Non-compute?
How many pair?

$$\mathbf{b} = \mathbf{G}^{\mathbf{b}}\mathbf{a}$$

$$\mathbf{a} = [0, 0, 1, 1]$$

$$(c\mathbf{I})(c\mathbf{I})(\jmath s \mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_4)(\jmath s \mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_4) = -s^2c^2\mathbf{Z}_2\mathbf{Z}_3$$

Non-Commute Pair Enumerator

- Number of non-commutable pairs

$$U(C,\gamma_1)^\dagger \mathbf{W}^{\mathbf{b}} U(C,\gamma_1) = U(C^{\mathbf{b}}, 2\gamma_1)^\dagger \mathbf{W}^{\mathbf{b}}$$

2^d Pauli-product terms: **b**=[0,0,..0] to [1,1,...,1]

$$U(C^{\mathbf{b}}, 2\gamma_1)^\dagger = \prod_{\nu}^{\rho} \mathrm{e}^{2\jmath\gamma_1 C_\nu} = \prod_{\nu}^{\rho} (c\mathbf{I} + \jmath s \prod_{\kappa} \mathbf{Z}_\kappa)$$

2^{rho} Pauli-product terms: **a**=[0,0,..0] to [1,1,...,1]

$$\mathbf{b} = \mathbf{G}^{\mathbf{b}} \mathbf{a}$$

$$\begin{bmatrix} 0 & 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad \mathbf{G}^{\mathbf{b}} = [\mathbf{G}]_{:,\{1,2,4,5\}} \quad \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 1 & \cdots & 1 \end{bmatrix}$$

Same?

$$\mathbf{b} = [1,1,0,1]$$

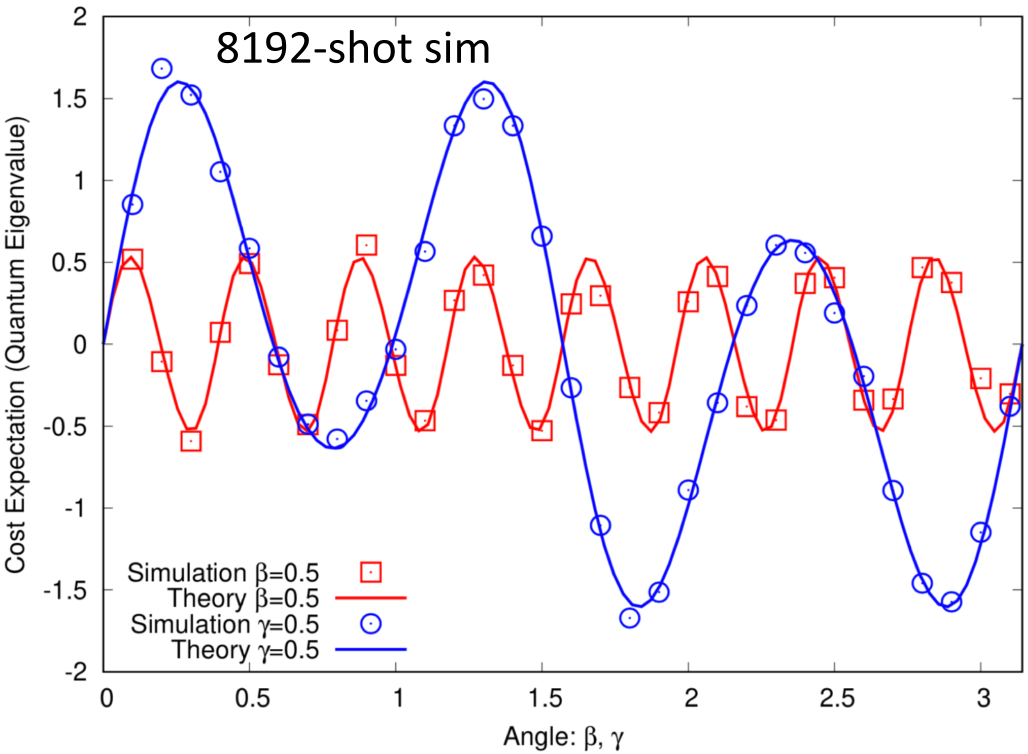
Counting the number of pairs: **A**

$$F_1(\gamma_1,\beta_1) = \sum_{\nu=1}^{n} (1-2y_\nu) \sum_{\mathbf{b}\in\mathbb{F}_2^k} \sum_{\mathbf{a}\in\mathbb{F}_2^\rho:\mathbf{b}=\mathbf{G}^{\mathbf{b}}\mathbf{a}} A_\nu^{\mathbf{a},\mathbf{b}} (\jmath s)^\omega c^{(\rho-\omega)} (-\jmath s')^\varpi c'^{(d_\nu^c-\varpi)}$$

Numerical Validation: Reed-Muller Code

- Corollary: QAOA-1 decoder of [16, 5] Reed-Muller codes has quantum eigenvalue:

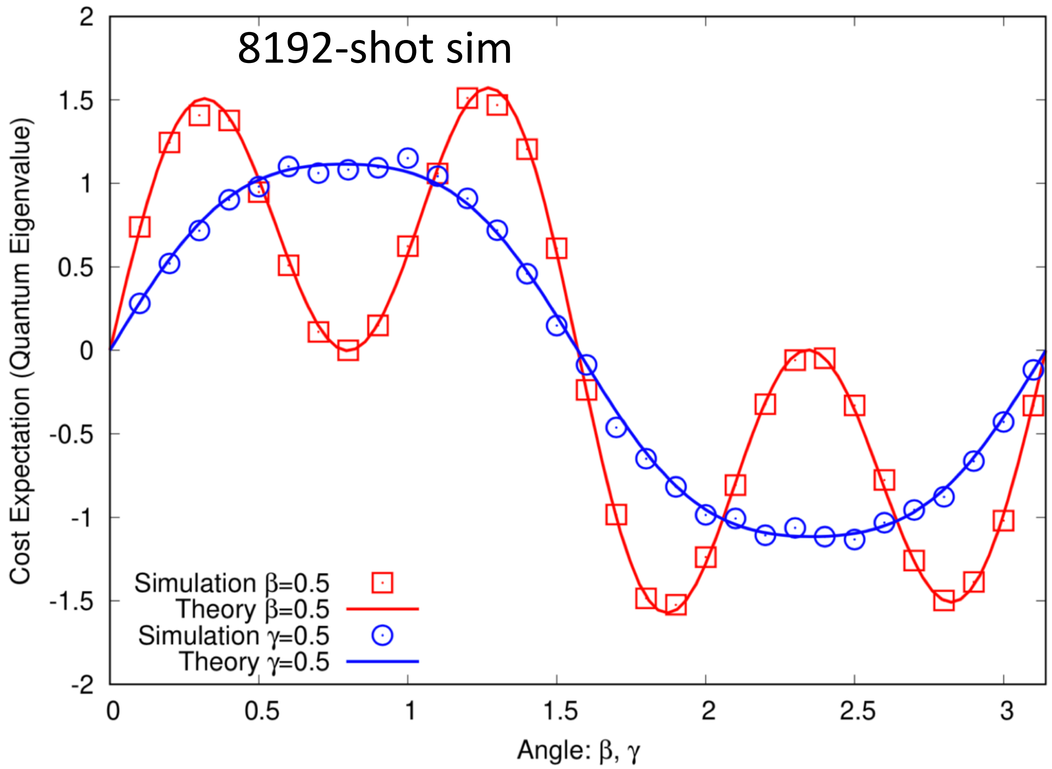
$$F_1(\gamma_1, \beta_1) = \frac{1}{32}\sin(4\gamma_1)\sin(2\beta_1)$$
$$(4(\cos(4\gamma_1) + \cos(12\gamma_1) + \cos(20\gamma_1) + \cos(24\gamma_1))\sin^4(2\beta_1)$$
$$+ 5(\cos(4\gamma_1) + \cos(12\gamma_1))(25 + 36\cos(4\beta_1) + 3\cos(8\beta_1))).$$



Numerical Validation: Hamming Code

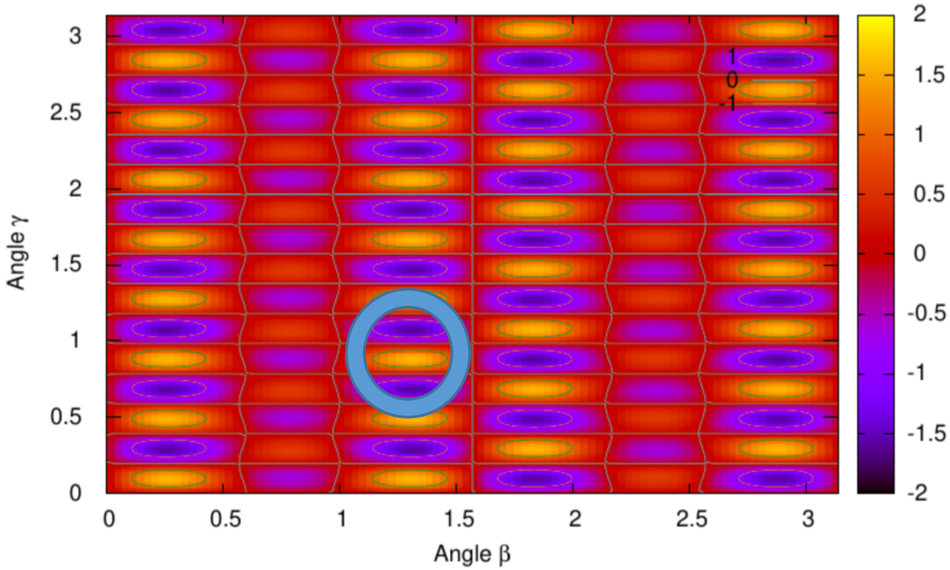
- Corrolay: QAOA-1 decoder of [7, 4] Hamming codes has quantum eigenvalue:

$$-2sc(c^2 - s^2)s'(1 - 3c'^2) + 3sc^2s'(1 + 2c'^2)$$

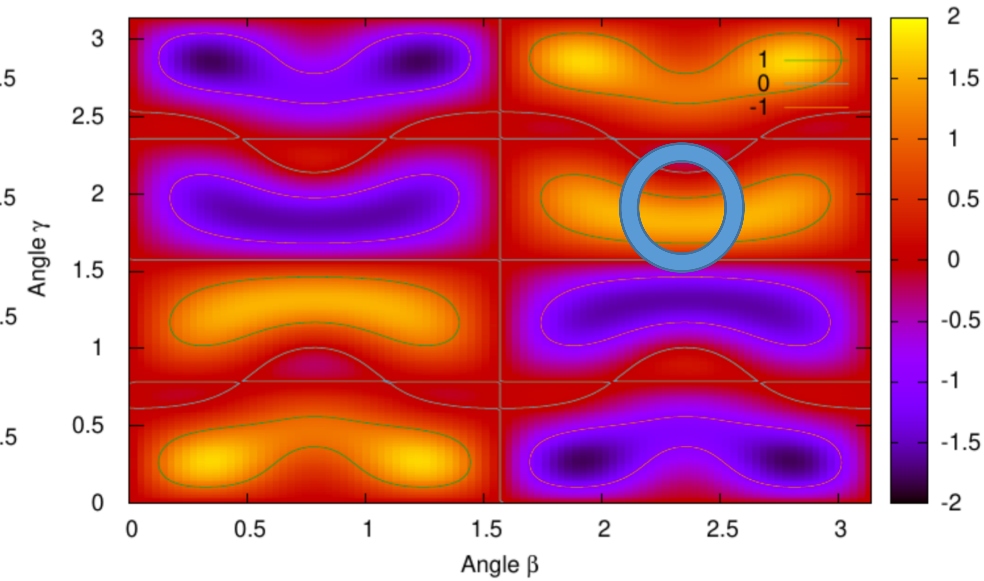


Variational Angle Optimization

- Given analytical expression of eigenvalues, we can obtain optimal variational angles without need of VQE



Reed-Muller Code



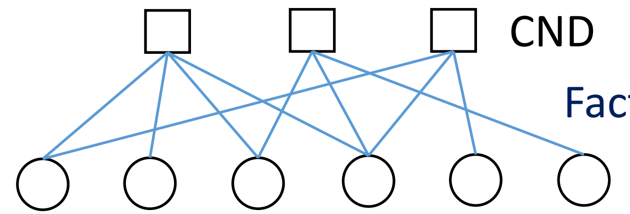
Hamming Code

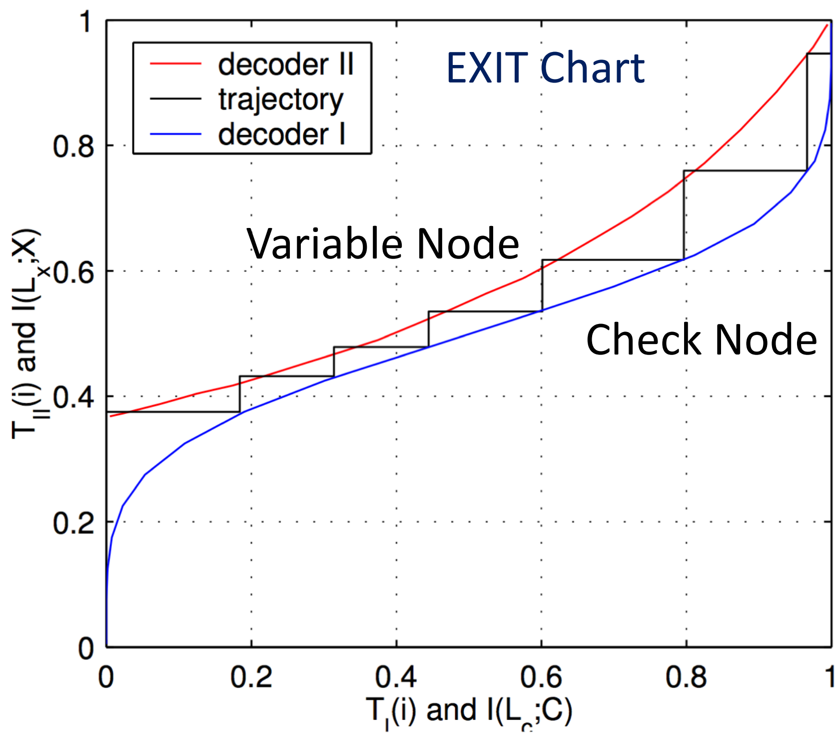
How To Optimize Codes for Quantum Decoder?

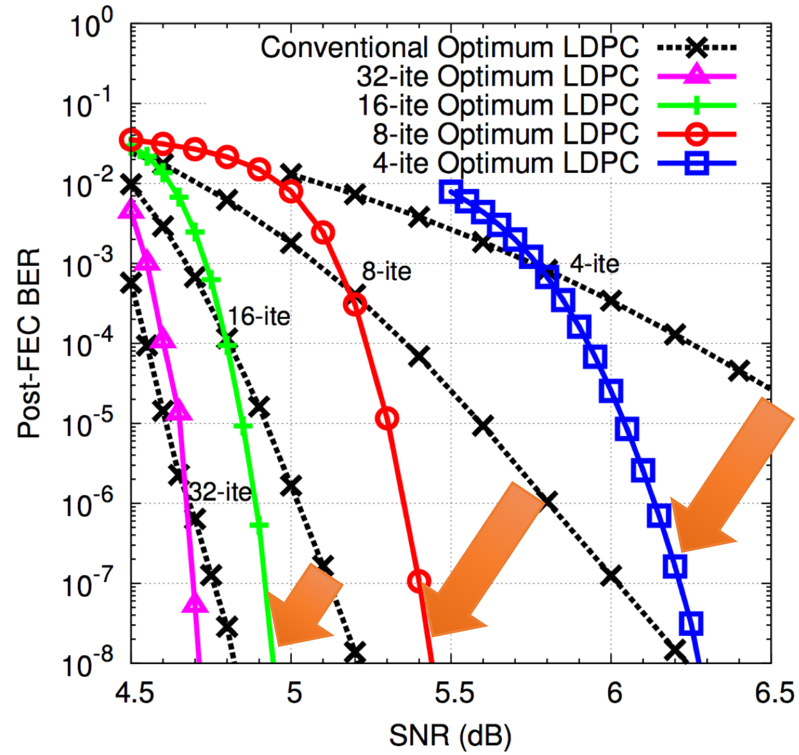
- It is known that degree distribution can be optimized by extrinsic information transfer (EXIT) or density evolution (DE) for LDPC codes when belief-propagation (BP) decoding is employed

Parity check

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$







Degree Distribution Analysis

- Any linear codes have exactly identical ML performance over arbitrary full-rank basis transform
  - Hamming distance spectrum is invariant
- Hamming code has average degree of 1.86, but it can be decreased to 1.71 and increased up to 2.71 via basis transform
- QAOA performance depends on degree distribution
  - Lower vs. higher degrees?

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{G}' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

d = [1 1 1 1 3 3 3]

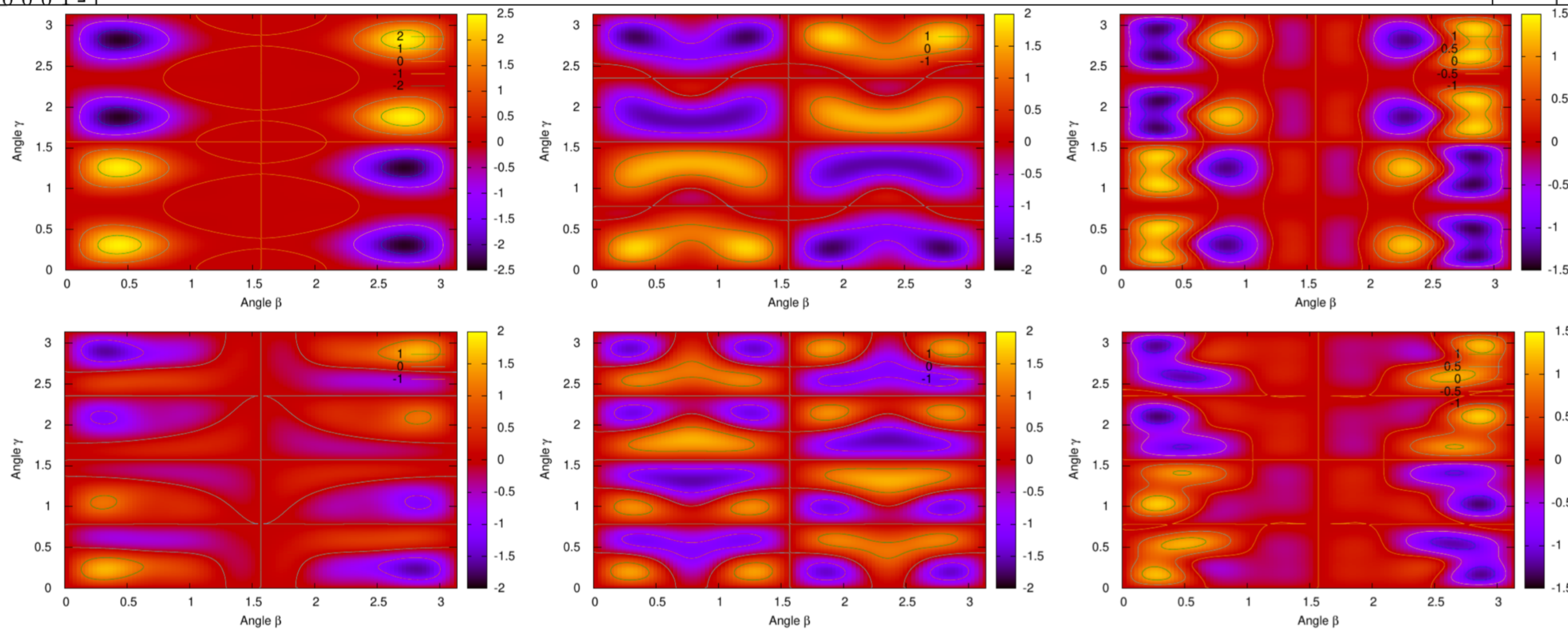
$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

d = [2 2 3 3 3 3 4]

$d_{\text{ave}} = 2.71$

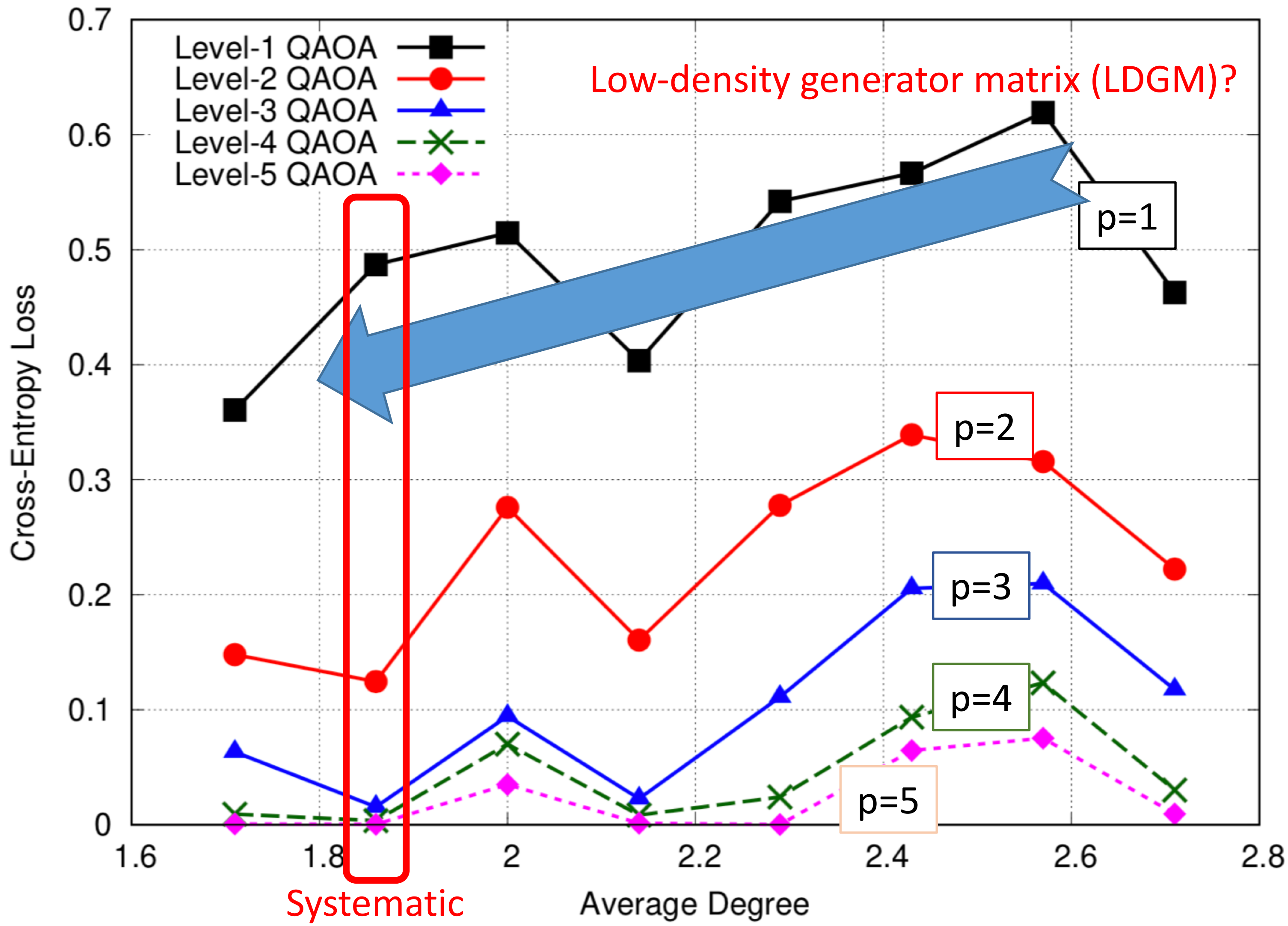
Transformed Hamming Codes: QAOA-1 Eigen

d^c	$\mathbf{P}$	$F_1(\gamma_1, \beta_1)$	$F_1^\star$	$\beta_1^\star$	$\gamma_1^\star$
1.71	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\1&0&0&1\end{bmatrix}$	$3sc^2s'(1+c')^2 - sc^2s'^3(c^2-3s^2)(c^2-s^2)$	2.409	0.424	0.311
1.86	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	$-2sc(c^2-s^2)s'(1-3c'^2) + 3sc^2s'(1+2c'^2)$	1.790	0.345	0.277
2.00	$\begin{bmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	$sc^2s'(1+c'+c'^2+3c'^3) + 2sc(c^2-s^2)s'(1+c'^2+2c'^3) - sc^2(c^2-3s^2)(c^2-s^2)s'^3(1+c')$	1.606	0.329	0.239
2.14	$\begin{bmatrix}1&1&0&0\\0&1&0&0\\0&1&1&0\\0&0&0&1\end{bmatrix}$	$3sc^2s'(c'^2-s'^2) + 2sc(c^2-s^2)s'(1+5c'^2)$	1.562	0.785	1.820
2.29	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\1&1&1&1\end{bmatrix}$	$-3sc^2s'(1-c'-3c'^2) + sc^2(c^2-3s^2)(c^2-s^2)s'(1+3c'+3c'^2)$	1.367	0.310	0.512
2.43	$\begin{bmatrix}1&1&1&1\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	$sc^2s'c'(1+2c'+3c'^2) - 2sc(c^2-s^2)s'(1+c'-2c'^2-2c'^3) + sc^2(c^2-3s^2)(c^2-s^2)s'(1+3c'+2c'^2+c'^3)$	1.308	0.283	1.034
2.57	$\begin{bmatrix}1&1&1&1\\0&1&1&1\\0&0&1&0\\0&0&0&1\end{bmatrix}$	$-sc^2s'(1+2c'-3c'^2-3c'^3) - 2sc(c^2-s^2)s'(1-3c'^2-2c'^3) + sc^2(c^2-3s^2)(c^2-s^2)s'(1+2c'+3c'^2+c'^3)$	1.420	0.275	1.005
2.71	$\begin{bmatrix}1&1&1&1\\0&1&1&1\\1&0&1&0\\0&0&0&1\end{bmatrix}$	$-3sc^2s'^3(1+c') + 2sc(c^2-s^2)s'c'(1+2c')(1+c') + sc^2(c^2-3s^2)(c^2-s^2)(3+3c'+c'^2)s'c'$	1.671	0.506	1.846



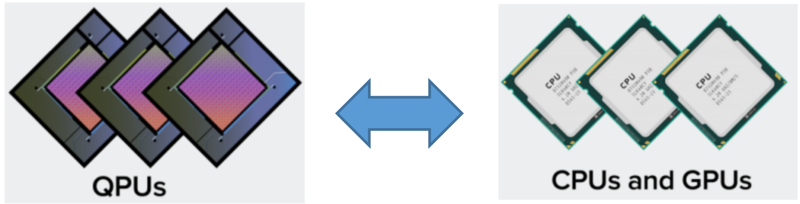
Transformed Hamming Codes: Level-p QAOA

- VQE with Nelder-Mead, Cross-entropy loss



Conclusions

- We introduced variational hybrid quantum-classical algorithms
- We applied QAOA to classical channel decoding problem
- We demonstrated the near-ML performance with QAOA decoding
- We evaluated performance on real quantum processor at IBM
- We developed theoretical framework to analyze QAOA decoding
- We optimized degree distribution of coding generator matrix focusing on Hamming codes ensemble
  - We observed that empirically LDGM works well for QAOA-1 decoding

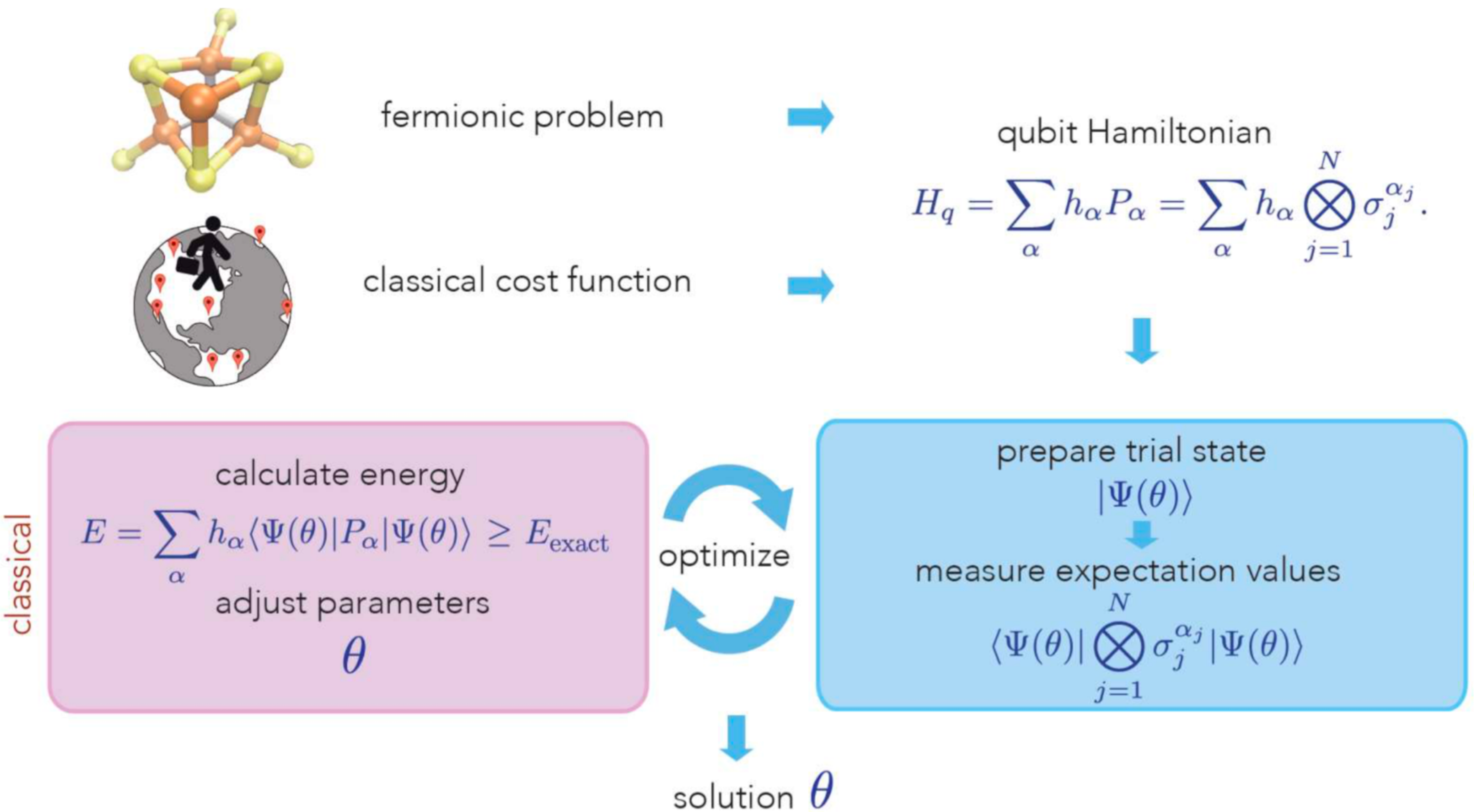


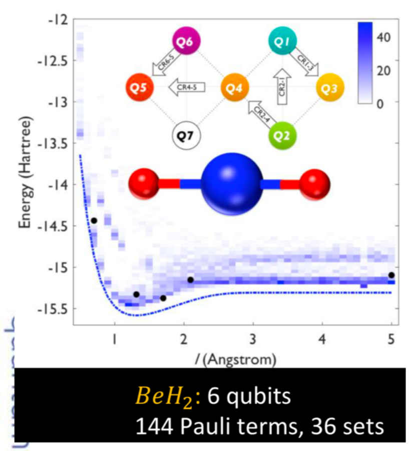
VQE: Variational Quantum Eigensolver

- Time evolution of quantum states: Schrodinger equation

$$H\left|\psi(t)\right\rangle = i\hbar\frac{\partial}{\partial t}\left|\psi(t)\right\rangle \qquad \left|\psi(t)\right\rangle = e^{-iHt/\hbar}\left|\psi(0)\right\rangle$$

- We obtain steady-eigenstates: $H\left|a\right\rangle = E_a\left|a\right\rangle$





Pauli Operator

- Single qubit over C^2 on sphere can be transformed by unitary operators (Stiefel Manifold), which has 4 degrees of freedom
- Complex 2x2 unitary operation is homomorphic to skew-Hermitian, decomposable by Pauli matrices

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \mathbf{Y} = \begin{bmatrix} 0 & -\jmath \\ \jmath & 0 \end{bmatrix}, \text{ and } \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

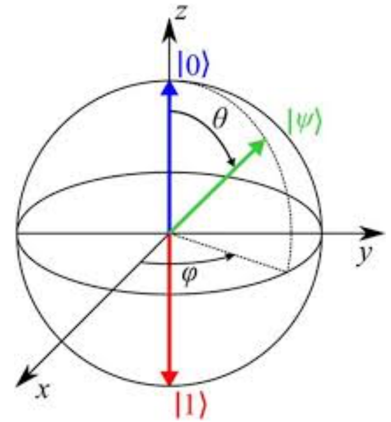
$$\mathbf{U} = \exp(\jmath(\beta_0 \mathbf{I} + \beta_1 \mathbf{X} + \beta_2 \mathbf{Y} + \beta_3 \mathbf{Z}))$$

- Exponential rule

$$\exp(\jmath\beta\mathbf{W}) = \cos(\beta)\mathbf{I} + \jmath\sin(\beta)\mathbf{W}$$

- Rotation rule

$$\mathbf{WW} = \mathbf{I}$$
$$\mathbf{XY} = \jmath\mathbf{Z}$$
$$\mathbf{YZ} = \jmath\mathbf{X}$$
$$\mathbf{ZX} = \jmath\mathbf{Y}$$



Commutation Rule Basics

- Mixer Hamiltonian operator is based on *X*-rotation (RX gates)
- Cost Hamiltonian operator is based on *Z*-rotation (RZ gates)

$$U(B,\beta) = \exp(-\jmath\beta B)$$

$$B = \sum_{\kappa=1}^{k} \mathbf{X}_\kappa$$

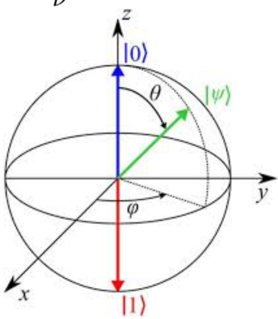
$$U(C,\gamma) = \exp(-\jmath\gamma C)$$

$$C = \sum_{\nu=1}^{n} C_\nu = \sum_{\nu=1}^{n} (1-2y_\nu) \prod_{\kappa\in\mathcal{I}_\nu^c} \mathbf{Z}_\kappa$$

- Quantum eigenvalue is conjugate product for cost function

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

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



- Commute or non-commute?

$$\exp(\jmath\beta\mathbf{W}) = \cos(\beta)\mathbf{I} + \jmath\sin(\beta)\mathbf{W}$$

$$\exp(-\jmath\beta\mathbf{Z})\mathbf{Z}\exp(\jmath\beta\mathbf{Z}) = \mathbf{Z}$$

Commute: No operation $[\mathbf{Z},\mathbf{Z}] = 0$

$$\exp(-\jmath\beta\mathbf{Z})\mathbf{Y}\exp(\jmath\beta\mathbf{Z}) = c\mathbf{Z} - s\mathbf{X}$$

Non-Commute: Rotation $[\mathbf{Z},\mathbf{Y}] = -2\jmath\mathbf{X}$

Entangled

$$\exp(-\jmath\beta\mathbf{Z}_1\mathbf{Z}_2)\mathbf{Y}_1\mathbf{Y}_2\exp(\jmath\beta\mathbf{Z}_1\mathbf{Z}_2) = \mathbf{Y}_1\mathbf{Y}_2$$

$[\mathbf{Z}_1\mathbf{Z}_2,\mathbf{Y}_1\mathbf{Y}_2] = 0$

Commute: No operation

Example: Reed-Muller Code

- Generator matrix

$$\begin{bmatrix} 1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1 \\ 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1 \\ 0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1 \\ 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 \\ 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1 \end{bmatrix}$$

- 10th Cost (degree d=3):

$$\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_5 \xrightarrow{\exp(\jmath\beta\mathbf{X})} (c'\mathbf{Z}_1 + s'\mathbf{Y}_1)(c'\mathbf{Z}_2 + s'\mathbf{Y}_2)(c'\mathbf{Z}_5 + s'\mathbf{Y}_5)$$

- Expansion: 2^d=8 terms with indicator **b**

b=[000]

$$c'^3\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3 + c'^2s'\mathbf{Z}_1\mathbf{Z}_2\mathbf{Y}_3 + c'^2s'\mathbf{Z}_1\mathbf{Y}_2\mathbf{Z}_3 + c's'^2\mathbf{Z}_1\mathbf{Y}_2\mathbf{Y}_3 +$$
$$c'^2s'\mathbf{Y}_1\mathbf{Z}_2\mathbf{Z}_3 + c's'^2\mathbf{Y}_1\mathbf{Z}_2\mathbf{Y}_3 + c's'^2\mathbf{Y}_1\mathbf{Y}_2\mathbf{Z}_3 + s'^3\mathbf{Y}_1\mathbf{Y}_2\mathbf{Y}_3$$

- Non-commute cost Hamiltonian of rank rho: $\mathbf{G}^{\mathbf{b}}$

b=[111]

$$\mathbf{bG}$$

$$(-\jmath s')^{\varpi} c'^{(d_\nu^{\mathrm{c}} - \varpi)}$$

- Expansion of cost Hamiltonian: 2^{rho} terms with indicator **a**

$$U(C^{\mathbf{b}}, 2\gamma_1)^\dagger = \prod_{\nu}^{\rho} \mathrm{e}^{2\jmath\gamma_1 C_\nu} = \prod_{\nu}^{\rho} (c\mathbf{I} + \jmath s \prod_{\kappa} \mathbf{Z}_\kappa)$$

$$(\jmath s)^{\omega} c^{(\rho - \omega)}$$

Example: Reed-Muller Code (Pair Counts)

- Generator matrix

$$\begin{bmatrix} 1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1 \\ 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1 \\ 0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1 \\ 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 \\ 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1 \end{bmatrix}$$

- Count non-commute pairs over **a** and **b** such that

$$\mathbf{b} = \mathbf{G}^{\mathbf{b}}\mathbf{a} \quad \Rightarrow \quad A_{\nu}^{\mathbf{a},\mathbf{b}} (\jmath s)^{\omega} c^{(\rho-\omega)} (-\jmath s')^{\varpi} c'^{(d_{\nu}^{c}-\varpi)}$$

- Enumerator results of RM codes:

```
A [   0    1    0   35    0  273    0  715    0  715    0  273    0   35    0    1    0] w 5 d 5 h 16
A [0 1 0 7 0 7 0 1 0] w 1 d 1 h 8     x 5
A [0 1 0 7 0 7 0 1 0] w 1 d 3 h 8     x 30
A [0 1 0 7 0 7 0 1 0] w 1 d 5 h 8     x 5
A [0 1 0 7 0 7 0 1 0] w 3 d 3 h 8     x 10
A [0 1 0 7 0 7 0 1 0] w 3 d 5 h 8     x 10
```

Side Note: Mean Cost vs. Cross Entropy

- VQE optimizes variational parameters based on Hamiltonian-energy relation:

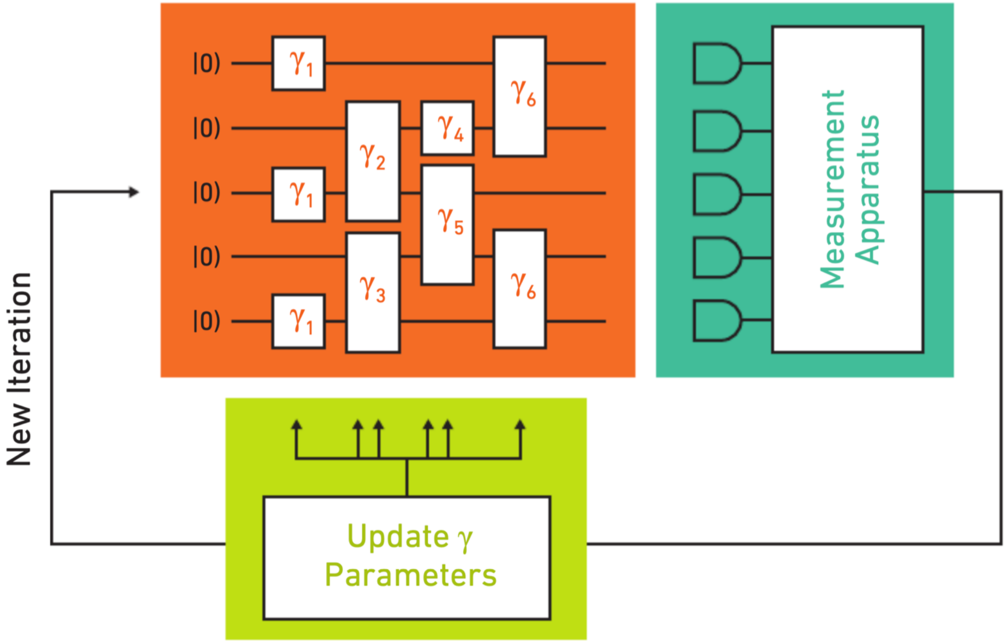
$$H\,|a\rangle = E_a\,|a\rangle$$

- For QAOA, we typically optimizer parameters to maximize **mean cost function**:

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

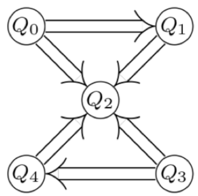
- It corresponds to minimizing average bit-error rate (BER)
- However, it does not minimize word-error rate (WER)
- We proposed to use cross entropy to minimize WER

$$\mathcal{H} = \mathbb{E} \log \sum_i \exp(-L_i)$$

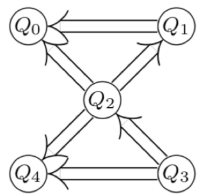


Side Note: Coupling Map for Real QPU

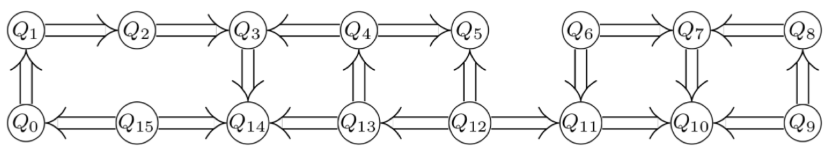
- We should be careful of the real quantum gate depths depending on QPU coupling maps
- CNOT bridging and SWAP should be reduced



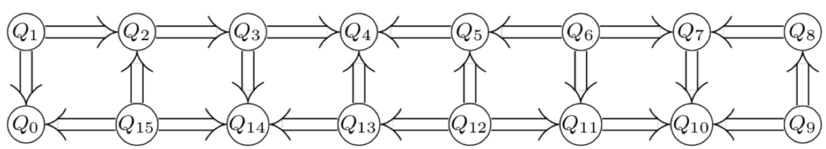
(a) *IBM QX2*



(b) *IBM QX4*

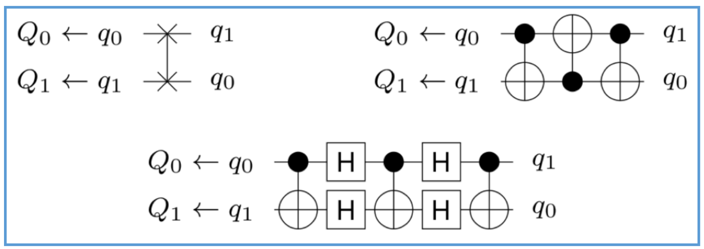


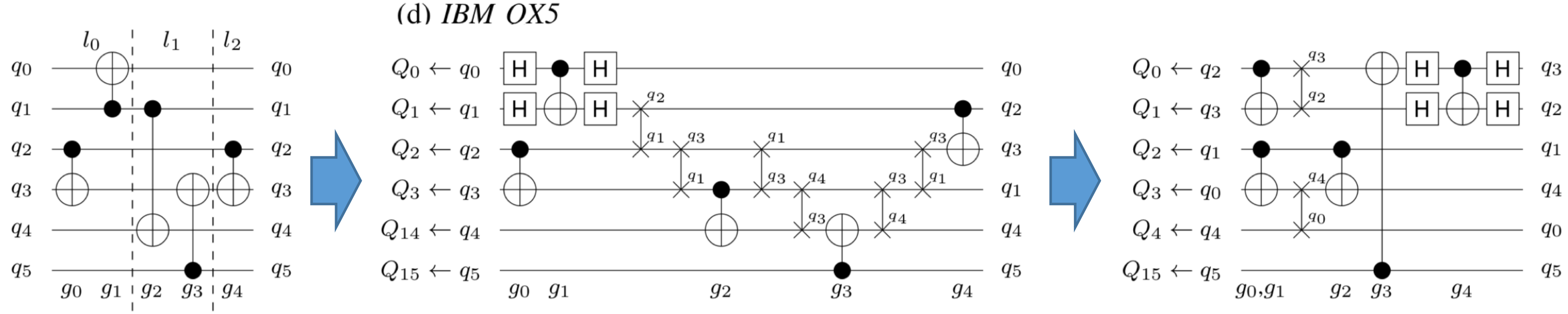
(c) *IBM QX3*



(d) *IBM QX5*

SWAP



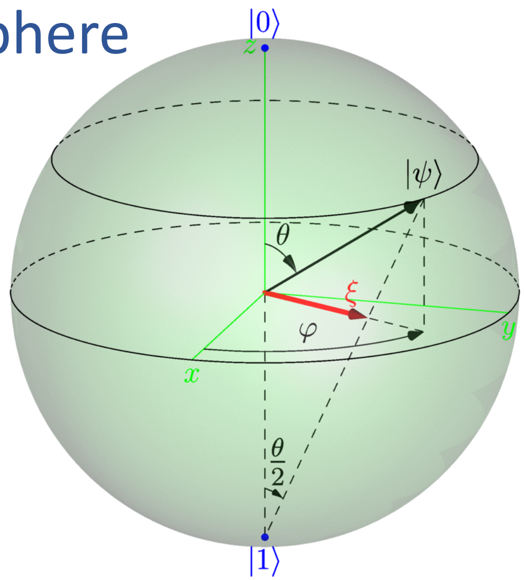


Bit vs. Qubit

- Classical bit: {0,1} → Quantum bit: superposition of |0> and |1>

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \qquad\qquad |\alpha_0|^2 + |\alpha_1|^2 = 1$$

Bloch Sphere



http://stla.github.io/stlapblog/posts/BlochSphere.html

THE BLOCH SPHERE

A stereographic representation of qubits

The simplest quantum state, namely the (pure) *qubit*, can be written

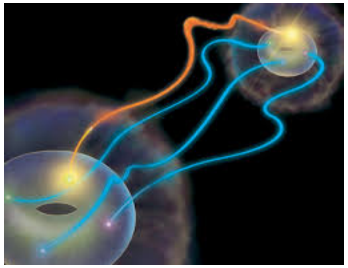
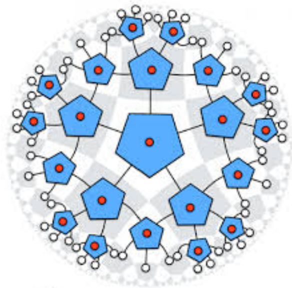
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

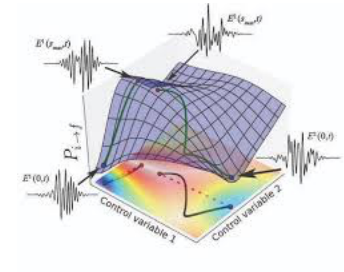
and shown on the *Bloch sphere* as the vector with spherical polar coordinates θ and φ. Of course, this representation of $|\psi\rangle$ on the sphere is *not* a linear combination of the representations of the basis states $|0\rangle$ and $|1\rangle$ at the poles of the sphere. However this graphical representation is not an artificial one. Indeed, taking the ratio of the two coordinates

$$\xi = \frac{e^{i\varphi}\sin\frac{\theta}{2}}{\cos\frac{\theta}{2}} = \tan\frac{\theta}{2}e^{i\varphi}$$

provides the *stereographic projection* of $|\psi\rangle$, which is shown in red on the picture. This ratio takes its value in the (xy)-plane *plus "a point at infinity"*, corresponding to the stereographic projection of $|1\rangle$. The other basis state $|0\rangle$ is sent to the origin of the (xy)-plane.

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \qquad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$



MITSUBISHI
ELECTRIC

Changes for the Better