



Complete guide to the Quantum Approximate Optimization Algorithm (QAOA)

Geilo Winter School

Franz G. Fuchs

19. - 24. January 2020

Quantum evolution

Postulate 2 [Nielsen and Chuang(2000), page 81]

The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\Phi\rangle$ of the system at time t_1 is related to the state $|\Phi'\rangle$ of the system at time t_2 by a **unitary operator** U which depends only on the times t_1 and t_2 ,

$$|\Phi'\rangle = U |\Phi\rangle \tag{1}$$

Let us derive the postulate.

Schrödinger Equation

The evolution of an isolated pure quantum state $|\Phi\rangle$ is described by the Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t} |\Phi(t)\rangle = H |\Phi(t)\rangle, \quad (2)$$

where H is the Hamiltonian of the system.

The Hamiltonian H is a Hermitian matrix, i.e., $H = H^\dagger$.

Schrödinger Equation

The evolution of an isolated pure quantum state $|\Phi\rangle$ is described by the Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t} |\Phi(t)\rangle = H |\Phi(t)\rangle, \quad (2)$$

where H is the Hamiltonian of the system.

The Hamiltonian H is a Hermitian matrix, i.e., $H = H^\dagger$. The solution is given by

$$|\Phi(t)\rangle = e^{-iHt} |\Phi(0)\rangle. \quad (3)$$

Schrödinger Equation

The evolution of an isolated pure quantum state $|\Phi\rangle$ is described by the Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t} |\Phi(t)\rangle = H |\Phi(t)\rangle, \quad (2)$$

where H is the Hamiltonian of the system.

The Hamiltonian H is a Hermitian matrix, i.e., $H = H^\dagger$. The solution is given by

$$|\Phi(t)\rangle = e^{-iHt} |\Phi(0)\rangle. \quad (3)$$

What does e to the power of a (Hermitian) matrix mean?

Exponentials of matrices

Exponential of a matrix defined through standard Taylor series

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \quad (4)$$

Exponentials of matrices

Exponential of a matrix defined through standard Taylor series

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \quad (4)$$

Example: Easy, if A is a diagonal matrix

$$e^{\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}} = \begin{pmatrix} e^{\lambda_1} & & \\ & \ddots & \\ & & e^{\lambda_n} \end{pmatrix} \quad (5)$$

Exponentials of matrices

Exponential of a matrix defined through standard Taylor series

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \quad (4)$$

Example: Easy, if A is a diagonal matrix

$$e^{\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}} = \begin{pmatrix} e^{\lambda_1} & & \\ & \ddots & \\ & & e^{\lambda_n} \end{pmatrix} \quad (5)$$

What about general Hermitian matrices?

Exponentials of Hermitian matrices

Theorem

For a Hermitian matrix $H \in \mathbb{C}^{n \times n}$ there exist n orthonormal eigenvectors and all eigenvalues are real. The matrix H admits the eigendecomposition

$$H = V\Lambda V^\dagger, \quad (6)$$

where the columns of V consist of the n orthonormal eigenvectors of A and the diagonal entries of Λ are given by the corresponding eigenvalues.

Exponentials of Hermitian matrices

Theorem

For a Hermitian matrix $H \in \mathbb{C}^{n \times n}$ there exist n orthonormal eigenvectors and all eigenvalues are real. The matrix H admits the eigendecomposition

$$H = V\Lambda V^\dagger, \quad (6)$$

where the columns of V consist of the n orthonormal eigenvectors of A and the diagonal entries of Λ are given by the corresponding eigenvalues.

The exponential of a Hermitian matrix H can be calculated as

$$e^H = \sum_{k=0}^{\infty} \frac{(V\Lambda V^\dagger)^k}{k!} = \sum_{k=0}^{\infty} \frac{V(\Lambda)^k V^\dagger}{k!} = V e^{\Lambda} V^\dagger \quad (7)$$

Quantum evolution is unitary.

Theorem

For a Hermitian matrix H and $t \in \mathbb{R}$, the matrix $U = e^{-iHt}$ is a unitary matrix, i.e., $UU^\dagger = U^\dagger U = I$.

Quantum evolution is unitary.

Theorem

For a Hermitian matrix H and $t \in \mathbb{R}$, the matrix $U = e^{-iHt}$ is a unitary matrix, i.e., $UU^\dagger = U^\dagger U = I$.

Proof: We have that

$$\begin{aligned} e^{-iHt} (e^{-iHt})^\dagger &= V e^{-i\Lambda t} V^\dagger \left(V e^{-i\Lambda t} V^\dagger \right)^\dagger = V e^{-i\Lambda t} V^\dagger V e^{i\Lambda t} V^\dagger = I, \\ (e^{-iHt})^\dagger e^{-iHt} &= \dots = I, \end{aligned} \tag{8}$$

Can we find a Hamiltonian for a given unitary matrix?

Theorem

Given a unitary matrix U we can always find a Hermitian matrix H such that $U = e^{-iHt}$.

Can we find a Hamiltonian for a given unitary matrix?

Theorem

Given a unitary matrix U we can always find a Hermitian matrix H such that $U = e^{-iHt}$.

- Diagonalize $U = VDV^\dagger$. For all j find λ_j such that $D_{jj} = e^{-i\lambda_j t}$.
- Not unique, since we can multiply with $e^{i2\pi k}$, $k \in \mathbb{Z}$.

Can we find a Hamiltonian for a given unitary matrix?

Theorem

Given a unitary matrix U we can always find a Hermitian matrix H such that $U = e^{-iHt}$.

- Diagonalize $U = VDV^\dagger$. For all j find λ_j such that $D_{jj} = e^{-i\lambda_j t}$.
- Not unique, since we can multiply with $e^{i2\pi k}$, $k \in \mathbb{Z}$.
- Note that, if $\lambda_i, |\Phi_i\rangle$ is an eigenpair of H , then $e^{-i\lambda_i t}, |\Phi_i\rangle$ is an eigenpair of $U = e^{-iHt}$.

Physics jargon

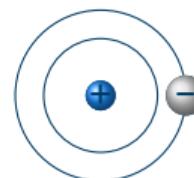
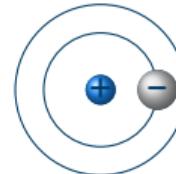
- Physicist call eigenvalues of a Hamiltonian for **energies**.
 - These values are amounts of energy the system can have.
 - They are all real and can be order from smallest to largest, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.
- To each energy λ_j corresponds to an **energy eigenstate**.
 - The energy eigenstate $|v_1\rangle$ corresponding to the lowest energy is called **ground state**.
 - The energy eigenstate $|v_2\rangle, |v_3\rangle, \dots$ are called **first excited state, second excited state, ...**

Physics jargon

- Physicist call eigenvalues of a Hamiltonian for **energies**.
 - These values are amounts of energy the system can have.
 - They are all real and can be order from smallest to largest, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.
- To each energy λ_j corresponds to an **energy eigenstate**.
 - The energy eigenstate $|v_1\rangle$ corresponding to the lowest energy is called **ground state**.
 - The energy eigenstate $|v_2\rangle, |v_3\rangle, \dots$ are called **first excited state, second excited state, ...**

Example:

Electron sitting in the lowest shell is in the ground state



First excited state has the electron in the next shell up

The adiabatic theorem

"A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum."

The adiabatic theorem

"A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum."

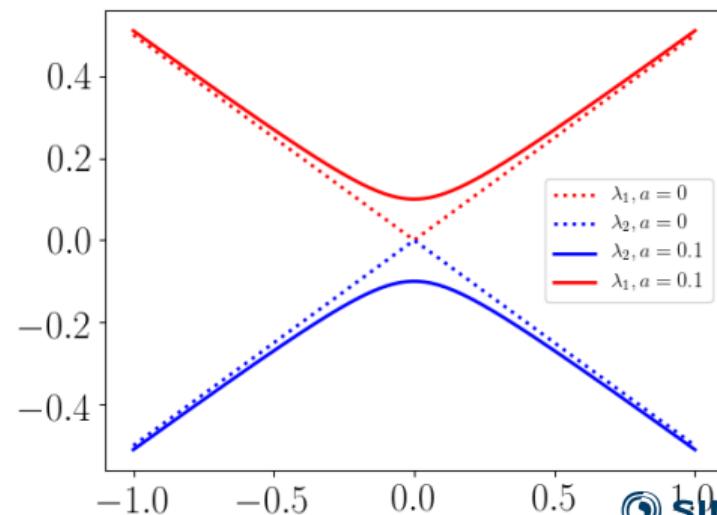
Consider a time dependent Hamiltonian

$$H(t) = \begin{pmatrix} \alpha t & a \\ a & -\alpha t \end{pmatrix} \quad (9)$$

$$\lambda_{1,2} = \pm \sqrt{a^2 + (\alpha t)^2} \quad (10)$$

The probability of a diabatic transition is given by (Landau-Zener)

$$P_D = e^{2\pi a^2 / |\alpha|} \quad (11)$$



Qauntum annealing

A predecessor of QAOA, quantum annealing (QA) has been widely studied for the purpose of solving combinatorial optimization problems. To find the MaxCut configuration that maximizes $\langle H_C \rangle$, we consider the following simple QA protocol:

$$H_{QA}(s) = -(sH_C + (1 - s)H_B), \quad s = t/T \tag{12}$$

- Ground state for $s = 0$ is $|+\rangle^{\otimes n}$.
- Ground state for $s = 1$ corresponds to solution encoded in H_C .

Methods to solve combinatorial optimization problems

- In **adiabatic QA**, the algorithm relies on the adiabatic theorem to remain in the instantaneous ground state along the annealing path, and solves the computational problem by finding the ground state at the end. To guarantee success, the necessary run time of the algorithm typically scales as $T = \mathcal{O}(1/\Delta_{\min}^2)$, where $\Delta_{\min} = \min_{s \in [0,1]} (\lambda_2(t) - \lambda_1(t))$ is the minimum spectral gap. It turns out that for hard instances, Δ_{\min} is exponentially small with respect to the problem size.

Methods to solve combinatorial optimization problems

- In **adiabatic QA**, the algorithm relies on the adiabatic theorem to remain in the instantaneous ground state along the annealing path, and solves the computational problem by finding the ground state at the end. To guarantee success, the necessary run time of the algorithm typically scales as $T = \mathcal{O}(1/\Delta_{\min}^2)$, where $\Delta_{\min} = \min_{s \in [0,1]} (\lambda_2(t) - \lambda_1(t))$ is the minimum spectral gap. It turns out that for hard instances, Δ_{\min} is exponentially small with respect to the problem size.
- Classical **simulated annealing** mimics adiabatic QA. But also takes exponential amount of time in the worst case.

Methods to solve combinatorial optimization problems

- In **adiabatic QA**, the algorithm relies on the adiabatic theorem to remain in the instantaneous ground state along the annealing path, and solves the computational problem by finding the ground state at the end. To guarantee success, the necessary run time of the algorithm typically scales as $T = \mathcal{O}(1/\Delta_{\min}^2)$, where $\Delta_{\min} = \min_{s \in [0,1]} (\lambda_2(t) - \lambda_1(t))$ is the minimum spectral gap. It turns out that for hard instances, Δ_{\min} is exponentially small with respect to the problem size.
- Classical **simulated annealing** mimics adiabatic QA. But also takes exponential amount of time in the worst case.
- The **adiabatic algorithm (QAOA)** can (at best) achieve Grover speedup.

Solving combinatorial optimization problems

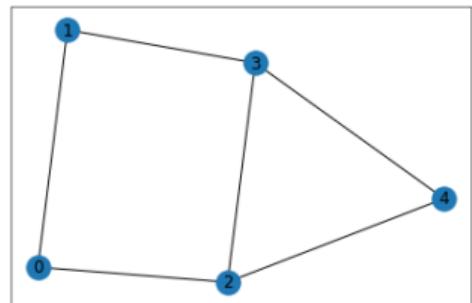
Example: weighted MAXCUT

- Given a graph $G = (V, E)$ consisting of vertices V and edges E with weights $w_{i,j} > 0$, for $(i,j) \in E$.
- A cut is defined as a partition of the vertices V into two disjoint subsets S, \bar{S} .
- The cost function to be maximized is the sum of weights of edges with vertices in the two different subsets.

Assign $x_i = \begin{cases} -1, & \text{if edge } i \text{ is in set } S \\ +1, & \text{otherwise} \end{cases}$, then the cost function is given by

$$C(x) = \sum_{(i,j) \in E} w_{i,j} \frac{1}{2} (1 - x_i x_j) \quad (13)$$

$$\begin{aligned} V &= \{0, 1, 2, 3, 4\} \\ E &= \{(0, 1, 1.0), (0, 2, 2.0), \\ &\quad (2, 3, 1.0), (3, 1, 2.0), \\ &\quad (3, 4, 1.0), (4, 2, 1.0)\} \end{aligned}$$



Types of approaches

Solving NP hard optimization problems.

- **Heuristic algorithms.** No polynomial run time guarantee; appear to perform well on some instances.
- **Approximate algorithms.** Efficient and provide provable guarantees.

Types of approaches

Solving NP hard optimization problems.

- **Heuristic algorithms.** No polynomial run time guarantee; appear to perform well on some instances.
- **Approximate algorithms.** Efficient and provide provable guarantees.
With high probability we get a solution x^* such that

$$\frac{\mathcal{C}(x^*)}{\max_x \mathcal{C}(x)} \geq \alpha, \quad (14)$$

where $0 < \alpha \leq 1$ is the approximation ratio.

Classical solution

- Calculating the cost of all partitions takes exponential time.

Classical solution

- Calculating the cost of all partitions takes exponential time.
- Polynomial time algorithm is randomized partitioning: for each edge $(i,j) \in E$ choose randomly S or \bar{S} with 50%. Therefore, the expectation value of a cut produced by random assignment can be written as follows:

$$\sum_{(i,j) \in E} w_{i,j} * Pr((i,j) \in \text{cut}) = \frac{1}{2} \sum_{e \in E} w_e \quad (15)$$

This produces a cut with expectation value of at least 0.5 times the maximum cut, since $\sum_{e \in E} w_e$ is an upper bound.

Classical solution

- Calculating the cost of all partitions takes exponential time.
- Polynomial time algorithm is randomized partitioning: for each edge $(i,j) \in E$ choose randomly S or \bar{S} with 50%. Therefore, the expectation value of a cut produced by random assignment can be written as follows:

$$\sum_{(i,j) \in E} w_{i,j} * Pr((i,j) \in \text{cut}) = \frac{1}{2} \sum_{e \in E} w_e \quad (15)$$

This produces a cut with expectation value of at least 0.5 times the maximum cut, since $\sum_{e \in E} w_e$ is an upper bound.

- Other polynomial approaches exist that involve semi-definite programming which give cuts of expected value at least 0.87856 times the maximum cut.

Express problem as ground state of Hamiltonian

- For each vertex we define $|x_i\rangle = \begin{cases} |0\rangle, & \text{if vertex } i \in S \\ |1\rangle, & \text{if vertex } i \in \bar{S} \end{cases}$
- Observe that

$$\begin{aligned}\sigma_z |0\rangle &= |0\rangle \\ \sigma_z |1\rangle &= -|1\rangle\end{aligned}\tag{16}$$

- The Hamiltonian encoding our problem is therefore

$$H_C = \sum_{(i,j) \in E} w_{i,j} \frac{1}{2} (I^n - I^a \otimes \sigma_z^i \otimes I^b \otimes \sigma_z^j \otimes I^c), \tag{17}$$

where I^m denotes the identity matrix in $(\mathbb{C}^2)^{\otimes m}$

Barbell example



Barbell example



Observe that, e.g.,

- $H_c |00\rangle = 1/2(I \otimes I - \sigma_z \otimes \sigma_z) |00\rangle = 1/2(|00\rangle - \sigma_z |0\rangle \otimes \sigma_z |0\rangle) = 1/2(|00\rangle - |0\rangle \otimes |0\rangle) = 0 |00\rangle$
- $H_c |01\rangle = 1/2(I \otimes I - \sigma_z \otimes \sigma_z) |01\rangle = 1/2(|01\rangle - \sigma_z |0\rangle \otimes \sigma_z |1\rangle) = 1/2(|01\rangle - |0\rangle \otimes (-|1\rangle)) = 1 |01\rangle$

Barbell example



Observe that, e.g.,

- $H_c |00\rangle = 1/2(I \otimes I - \sigma_z \otimes \sigma_z) |00\rangle = 1/2(|00\rangle - \sigma_z |0\rangle \otimes \sigma_z |0\rangle) = 1/2(|00\rangle - |0\rangle \otimes |0\rangle) = 0 |00\rangle$
- $H_c |01\rangle = 1/2(I \otimes I - \sigma_z \otimes \sigma_z) |01\rangle = 1/2(|01\rangle - \sigma_z |0\rangle \otimes \sigma_z |1\rangle) = 1/2(|01\rangle - |0\rangle \otimes (-|1\rangle)) = 1 |01\rangle$

This means that

- $|00\rangle$, and $|11\rangle$ are eigenkets of $-H_C$ with eigenvalue 0.
- $|01\rangle$, and $|10\rangle$ are eigenkets of $-H_C$ with eigenvalue -1.

(weighted) Max-Cut

$$H_C = \sum_{(j,k) \in E} \frac{1}{2} w_{i,j} \left(I - \sigma_z^i \sigma_z^j \right) \quad (18)$$

- H_C is sum of $|E|$ local terms
- H_C is a diagonal matrix

(weighted) Max-Cut

$$H_C = \sum_{(j,k) \in E} \frac{1}{2} w_{i,j} \left(I - \sigma_z^i \sigma_z^j \right) \quad (18)$$

- H_C is sum of $|E|$ local terms
- H_C is a diagonal matrix

$$H_B = \sum_{i \in \text{nodes}} \sigma_x^i \quad (19)$$

- H_B has only off-diagonal non-zero entries
- H_B induces a swap operation between neighboring qubits, and thus can move the excitation around for the purpose of state transfer

How to find quantum gates for QA?

We need to find gates for

$$e^{-iH_{QA}(s)}, \quad (20)$$

where

$$H_{QA}(s) = -(sH_C + (1 - s)H_B), \quad s = t/T \quad (21)$$

Adding of Hamiltonians

If H_1, H_2 are matrices (Hamiltonians), then

$$e^{H_1+H_2} \neq e^{H_1}e^{H_2}, \quad (22)$$

except when H_1 and H_2 commute, i.e., $H_1H_2 = H_2H_1$.

Adding of Hamiltonians

If H_1, H_2 are matrices (Hamiltonians), then

$$e^{H_1+H_2} \neq e^{H_1}e^{H_2}, \quad (22)$$

except when H_1 and H_2 commute, i.e., $H_1H_2 = H_2H_1$.

Trotterization, (Lie-Trotter-Suzuki product formula[Trotter(1959), Suzuki(1976)])

$$e^{-i(H_1+H_2)t} = \left(e^{-iH_1\frac{t}{n}}e^{-iH_2\frac{t}{n}}\right)^n + \mathcal{O}\left(\frac{t^2}{n}\right) \quad (23)$$

First and second order versions

$$\begin{aligned} e^{-i(H_1+H_2)t} &= e^{-iH_1t}e^{-iH_2t} + \mathcal{O}(t^2) \\ e^{-i(H_1+H_2)t} &= e^{-iH_1t/2}e^{-iH_2t}e^{-iH_1t/2} + \mathcal{O}(t^3) \end{aligned} \quad (24)$$

Overall QAOA

1. Using $2p$ parameters $\gamma = \gamma_1, \dots, \gamma_p, \beta = \beta_1, \dots, \beta_p$, prepare state

$$|\Psi(\gamma, \beta)\rangle = U_{B_p} U_{C_p} \dots U_{B_1} U_{C_1} |+\rangle^{\otimes n}, \quad (25)$$

where the operators have the explicit form

$$\begin{aligned} U_{B_l} &= e^{-i\beta_l H_B} = \prod_{j=1}^n e^{-i\beta_l \sigma_x^j}, \\ U_{C_l} &= e^{-i\beta_l H_C} = \prod_{(j,k) \in E} e^{-i\gamma_l w_{j,k}/2(I - \sigma_z^j \sigma_z^k)}, \end{aligned} \quad (26)$$

2. Obtain $\langle \Psi(\gamma, \beta) | H_C | \Psi(\gamma, \beta) \rangle$.
3. Run an outer, classical, optimization loop to find γ, β that minimizes the expectation value $\langle \Psi(\gamma, \beta) | H_C | \Psi(\gamma, \beta) \rangle$.

How to obtain the expectation value

H_C is a diagonal Hamiltonian, and we have that

$$H_C = \sum_{x \in \{0,1\}^n} c(x) |x\rangle\langle x| \quad (27)$$

Therefore,

$$\begin{aligned} \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | H | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle &= \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | \sum_{x \in \{0,1\}^n} c(x) |x\rangle\langle x| | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle \\ &= \sum_{x \in \{0,1\}^n} c(x) \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | x \rangle \langle x | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle = \sum_{x \in \{0,1\}^n} c(x) p(x) \end{aligned} \quad (28)$$

How to obtain the expectation value

H_C is a diagonal Hamiltonian, and we have that

$$H_C = \sum_{x \in \{0,1\}^n} c(x) |x\rangle\langle x| \quad (27)$$

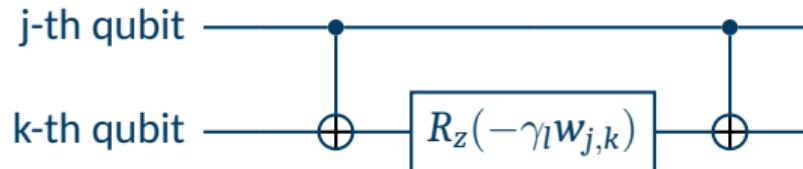
Therefore,

$$\begin{aligned} \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | H | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle &= \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | \sum_{x \in \{0,1\}^n} c(x) |x\rangle\langle x| | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle \\ &= \sum_{x \in \{0,1\}^n} c(x) \langle \Psi_p(\vec{\gamma}, \vec{\beta}) | x \rangle \langle x | \Psi_p(\vec{\alpha}, \vec{\beta}) \rangle = \sum_{x \in \{0,1\}^n} c(x) p(x) \end{aligned} \quad (28)$$

Remember that, given a random outcome x' , we only need to calculate the cost function once.

How to implement with gates efficiently?

$e^{-i\gamma_l w_{j,k}/2(I - \sigma_z^j \sigma_z^k)}$ can be implemented as



- Observe that $e^{-i\gamma_l w_{j,k}/2I}$ is a global phase and can be ignored
-

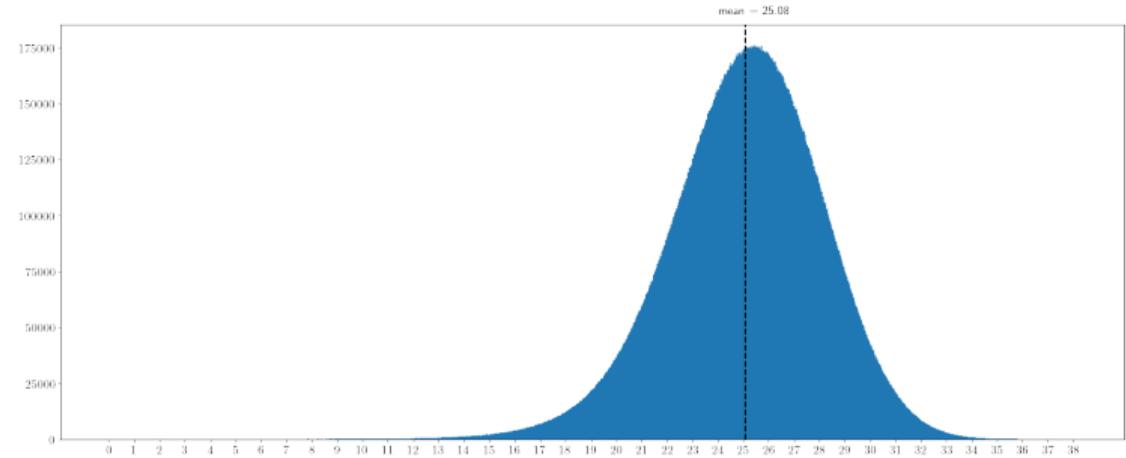
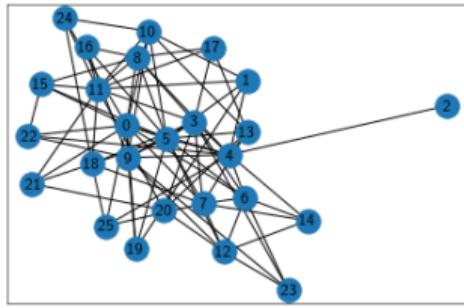
$$\begin{aligned} (CX)(I \otimes Rz(\theta))(CX) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} e^{-i\theta/2} & 0 & 0 & 0 \\ 0 & e^{i\theta/2} & 0 & 0 \\ 0 & 0 & e^{i\theta/2} & 0 \\ 0 & 0 & 0 & e^{-i\theta/2} \end{pmatrix} = e^{-i\theta/2\sigma_z\sigma_z} \end{aligned}$$

How to implement with gates efficiently?

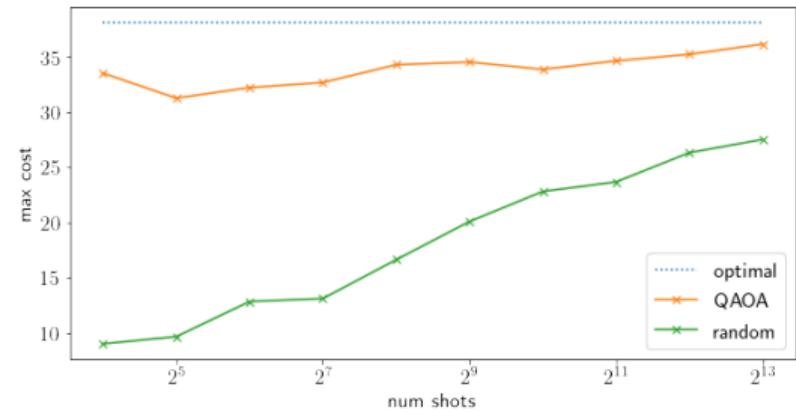
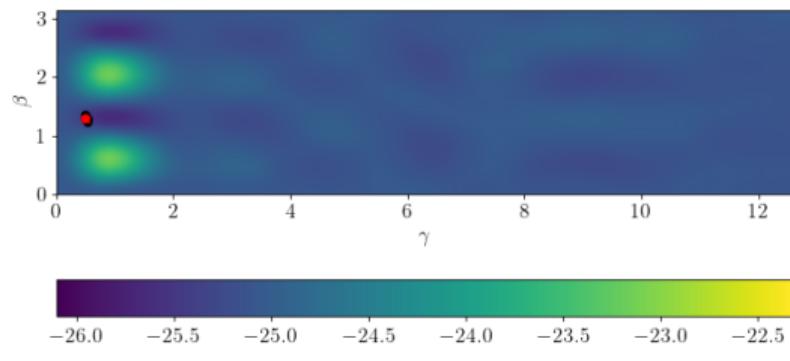
$e^{-i\beta_l \sigma_x^j}$ can be implemented as j-th qubit ————— $R_x(2\beta_l)$ —————

$$Rx(\theta) = \begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \quad (30)$$

Example



Example





Technology for a better society

-  M.A. Nielsen and I.L. Chuang.
Quantum Computation and Quantum Information.
Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.
-  Masuo Suzuki.
Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems.
Communications in Mathematical Physics, 51(2):183–190, 1976.
-  Hale F Trotter.
On the product of semi-groups of operators.
Proceedings of the American Mathematical Society, 10(4):545–551, 1959.