Variational Quantum Eigensolver and Quantum Approximate Optimization Algorithm

Fabian Köhler

23.11.2021

University of Hamburg

Motivation

- quantum mechanics has unique features (superposition of states, entanglement, ...)
- exploit these features to perform computations
- suggest some advantages over classical computers (solve certain problems substantially faster)

Motivation

- quantum mechanics has unique features (superposition of states, entanglement, ...)
- exploit these features to perform computations
- suggest some advantages over classical computers (solve certain problems substantially faster)
- different quantum computing paradigms have been proposed
  - digital quantum computer (gate-based approach)
  - adiabatic quantum computing and quantum annealing (analog approaches)
  - quantum cellular automata
  - ...

Motivation

- quantum mechanics has unique features (superposition of states, entanglement, ...)
- exploit these features to perform computations
- suggest some advantages over classical computers (solve certain problems substantially faster)
- different quantum computing paradigms have been proposed
  - digital quantum computer (gate-based approach)
  - adiabatic quantum computing and quantum annealing (analog approaches)
  - quantum cellular automata
  - ...
- limited usefulness due to hardware constraints

Motivation

- quantum mechanics has unique features (superposition of states, entanglement, ...)
- exploit these features to perform computations
- suggest some advantages over classical computers (solve certain problems substantially faster)
- different quantum computing paradigms have been proposed
  - digital quantum computer (gate-based approach)
  - adiabatic quantum computing and quantum annealing (analog approaches)
  - quantum cellular automata
  - ...
- limited usefulness due to hardware constraints
- hybrid approaches:
  - variational quantum eigensolver and quantum approximate optimization algorithm
  - “robust” against gate imperfections, shallow circuits, already useful for a limited number of qubits

Notation

- notation for the Pauli matrices

$$I = \sigma^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad X = \sigma^1 = \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1}$$

$$Y = \sigma^2 = \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \sigma^3 = \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2}$$

- subscripts indicate qubit: $X_1 Y_2 = X \otimes Y$
- computational basis is z-basis:

$$|0\rangle = |\downarrow\rangle \tag{3}$$

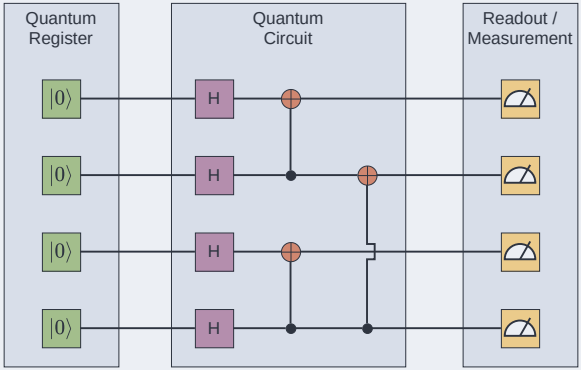
$$|1\rangle = |\uparrow\rangle \tag{4}$$

Table of Contents

Digital Quantum Computer

Digital QC: Principle

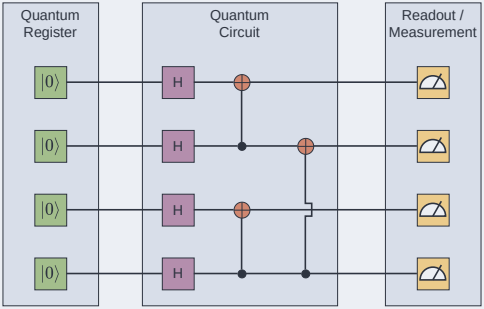
- applies quantum gates to registers of qubits, performs arbitrary computations
- interesting algorithms (e.g. Shor or Grover) have been proposed to solve problems faster than on a classical computer



Digital QC: Vincenzo's Criteria

Criteria for a universal quantum computer

- well-defined qubits (two-level system)
- initialization of the qubits in a well defined-state
- long decoherence time
- a universal set of quantum gates (very small set of one- and two-qubit gates)
- qubit-selective readout



Digital QC: Qubit Implementations

- many platforms have been suggested/explored:
  - Josephson junctions
  - photonic qubits
  - Rydberg atoms
  - electrons
  - nuclear spin (NMR)
  - quantum dots
  - topological systems (non-abelian anyons)
  - optical lattices (internal atomic states)
  - van der Waals heterostructures
  - ...

Digital QC: Qubit Implementations

- many platforms have been suggested/explored:
  - Josephson junctions
  - photonic qubits
  - Rydberg atoms
  - electrons
  - nuclear spin (NMR)
  - quantum dots
  - topological systems (non-abelian anyons)
  - optical lattices (internal atomic states)
  - van der Waals heterostructures
  - ...
- platforms differ vastly regarding the relevant characteristics
  - gate fidelities
  - connectivity
  - storage and decoherence time
  - scalability (number of qubits)

Digital QC: Characteristics

Advantages:

- can run arbitrary quantum algorithms (quantum Turing machine)

Digital QC: Characteristics

Advantages:

- can run arbitrary quantum algorithms (quantum Turing machine)

Disadvantages:

- limited by currently available hardware
  - limited number of gate operations due to short decoherence times and gate infidelity
  - number of qubits is generally small
  - quantum error correction demands even more qubits

Digital QC: Characteristics

Advantages:

- can run arbitrary quantum algorithms (quantum Turing machine)

Disadvantages:

- limited by currently available hardware
  - limited number of gate operations due to short decoherence times and gate infidelity
  - number of qubits is generally small
  - quantum error correction demands even more qubits
- number of proposed algorithms is fairly limited

Digital QC: Characteristics

Advantages:

- can run arbitrary quantum algorithms (quantum Turing machine)

Disadvantages:

- limited by currently available hardware
  - limited number of gate operations due to short decoherence times and gate infidelity
  - number of qubits is generally small
  - quantum error correction demands even more qubits
- number of proposed algorithms is fairly limited
- so far only one claim of actual quantum advantage:
  - Google 2019: sampling random quantum circuits
  - 53 superconducting qubits
  - classical supercomputer was limited to 43 qubits

Analog Quantum Computer

Analog QC: Principle I

Here: Analog Quantum Computing = Adiabatic Quantum Computing

- solve an optimization problem encoded in a problem Hamiltonian H_{P}
- start with the ground state of a well known Hamiltonian

$$H_0 = \sum_{i=1}^{N} X_i \tag{5}$$

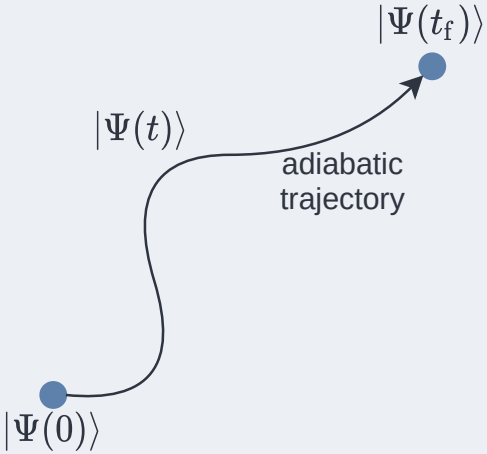
- slowly drive the Hamiltonian to H_{P}

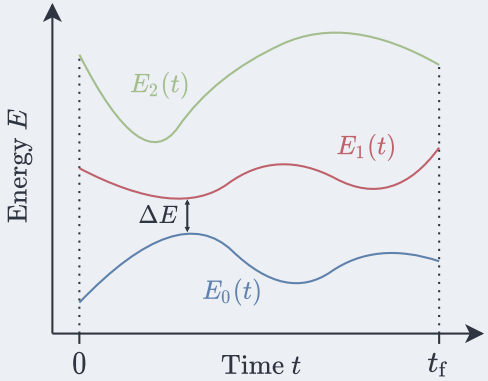
$$H(t) = \big(1 - f(t)\big)H_0 + f(t)H_{\mathrm{P}}, \quad f(0) = 0, \quad f(t_{\mathrm{f}}) = 1 \tag{6}$$

- if done slow enough the system will remain in the instantaneous ground state (adiabatic theorem)
- maximum feasible speed depends on minimum energy gap in the system

$$t_{\mathrm{f}} = \mathcal{O}(\Delta E^{-2}) \tag{7}$$

Analog QC: Principle II





Characteristics

Advantages:

- well suited for optimization/eigenvalue problems
- commercial platforms with thousands of qubits available (e.g. D-Wave with superconducting qubits)

Characteristics

Advantages:

- well suited for optimization/eigenvalue problems
- commercial platforms with thousands of qubits available (e.g. D-Wave with superconducting qubits)

Disadvantages:

- not universal, at least on current hardware (e.g. Shor algorithm cannot be mapped)
- near degeneracies/avoided crossings require slow driving
- no evidence for quantum speed-up so far

Variational Quantum Eigensolver (VQE)

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_C(\vec{\theta})$)

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_{\mathrm{C}}(\vec{\theta})$)
  2. evaluate the cost function (i.e. measure some observable) w.r.t. this trial state

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_{\mathrm{C}}(\vec{\theta})$)
  2. evaluate the cost function (i.e. measure some observable) w.r.t. this trial state
  3. employ a classical optimizer to propose a new set of parameters

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_C(\vec{\theta})$)
  2. evaluate the cost function (i.e. measure some observable) w.r.t. this trial state
  3. employ a classical optimizer to propose a new set of parameters
  4. repeat until converged

VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_{\mathrm{C}}(\vec{\theta})$)
  2. evaluate the cost function (i.e. measure some observable) w.r.t. this trial state
  3. employ a classical optimizer to propose a new set of parameters
  4. repeat until converged
- variational principle guarantees

$$\langle\Psi(\vec{\theta})|H_{\mathrm{P}}|\Psi(\vec{\theta})\rangle \geq E_0 \tag{8}$$

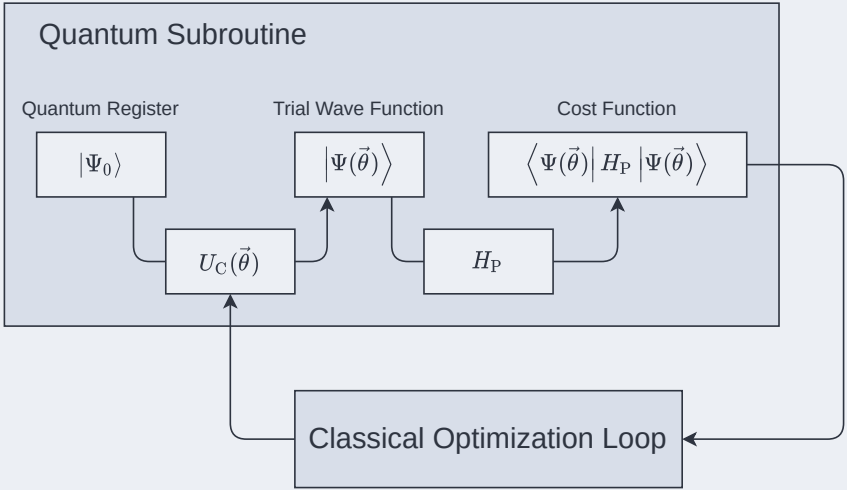
VQE: Principle

- hybrid approaches that render today's quantum hardware more useful
- solve optimization/eigenvalue problems
- combine a quantum computer with a classical optimizer
  1. prepare a state $|\Psi(\vec{\theta})\rangle$ according to some parameters $\vec{\theta}$ (e.g. parametric circuit $U_{\mathrm{C}}(\vec{\theta})$)
  2. evaluate the cost function (i.e. measure some observable) w.r.t. this trial state
  3. employ a classical optimizer to propose a new set of parameters
  4. repeat until converged
- variational principle guarantees

$$\langle\Psi(\vec{\theta})|H_{\mathrm{P}}|\Psi(\vec{\theta})\rangle \geq E_0 \tag{8}$$

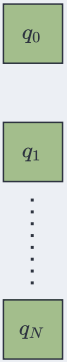
- the optimization of parameters can mitigate imperfections of the hardware to some extent

VQE: Feedback Loop



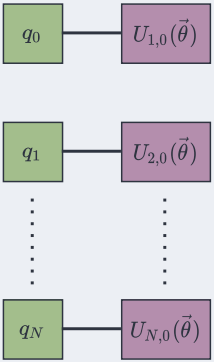
VQE: Typical Structure of a Parametric Circuit

N qubits store the trial states, each encoding a classical bit



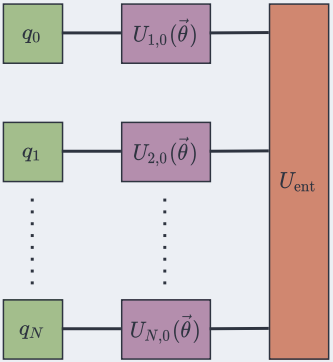
VQE: Typical Structure of a Parametric Circuit

single qubit gates rotate each qubit according to some parameters $\vec{\theta}$



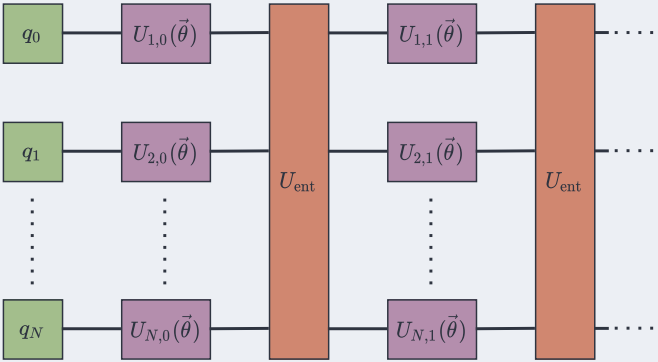
VQE: Typical Structure of a Parametric Circuit

entanglers generate entanglement among the qubits, crucial for quantum speed-up

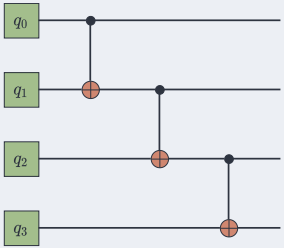


VQE: Typical Structure of a Parametric Circuit

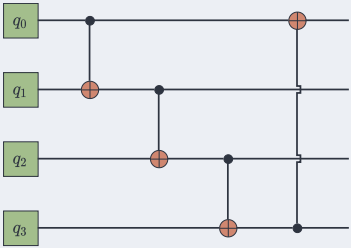
operator structure is repeated



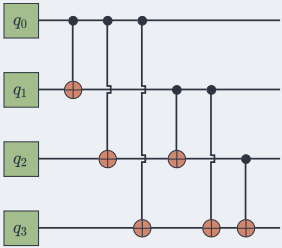
VQE: Entanglement Schemes



linear



circular



full

- of course other schemes are conceivable as well
- best choice depends on the connectivity of the hardware platform and the problem under consideration
- it would be nice to have a simple scheme guiding the construction of such circuits

VQE: Molecular Hamiltonian

- electronic problem in Born-Oppenheimer approximation

$$H = \underbrace{-\sum_{i=1}^{N_e} \frac{\hbar^2}{2m_e} \nabla^2}_{\text{kinetic}} \underbrace{-\sum_{i=1}^{N_n} \sum_{j=1}^{N_e} \frac{Z_i e^2}{4\pi\varepsilon_0 \left|\vec{R}_i - \vec{r}_j\right|}}_{\text{electron-nuclei int.}} + \underbrace{\frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_e} \frac{e^2}{4\pi\varepsilon_0 \left|\vec{r}_i - \vec{r}_j\right|}}_{\text{electron-electron int.}} \tag{9}$$

VQE: Molecular Hamiltonian

- electronic problem in Born-Oppenheimer approximation

$$H = \underbrace{-\sum_{i=1}^{N_e} \frac{\hbar^2}{2m_e} \nabla^2}_{\text{kinetic}} \underbrace{-\sum_{i=1}^{N_n} \sum_{j=1}^{N_e} \frac{Z_i e^2}{4\pi\varepsilon_0 \left|\vec{R}_i - \vec{r}_j\right|}}_{\text{electron-nuclei int.}} + \underbrace{\frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_e} \frac{e^2}{4\pi\varepsilon_0 \left|\vec{r}_i - \vec{r}_j\right|}}_{\text{electron-electron int.}} \tag{9}$$

- Hartree-Fock method allows for efficient mean-field treatment
- limited usefulness, does not incorporate electronic correlations

VQE: Molecular Hamiltonian

- electronic problem in Born-Oppenheimer approximation

$$H = \underbrace{-\sum_{i=1}^{N_e} \frac{\hbar^2}{2m_e} \nabla^2}_{\text{kinetic}} \underbrace{-\sum_{i=1}^{N_n} \sum_{j=1}^{N_e} \frac{Z_i e^2}{4\pi\varepsilon_0 \left|\vec{R}_i - \vec{r}_j\right|}}_{\text{electron-nuclei int.}} + \underbrace{\frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_e} \frac{e^2}{4\pi\varepsilon_0 \left|\vec{r}_i - \vec{r}_j\right|}}_{\text{electron-electron int.}} \tag{9}$$

- Hartree-Fock method allows for efficient mean-field treatment
- limited usefulness, does not incorporate electronic correlations
- can serve as a starting point for post-Hartree-Fock methods like configuration interaction or coupled cluster
- these methods are computationally hard due to the scaling w.r.t. system size
- only feasible for “small” molecules, typically up to $\mathcal{O}(10^2)$ electrons

VQE: Molecular Hamiltonian

- electronic problem in Born-Oppenheimer approximation

$$H = \underbrace{-\sum_{i=1}^{N_e} \frac{\hbar^2}{2m_e} \nabla^2}_{\text{kinetic}} \underbrace{-\sum_{i=1}^{N_n} \sum_{j=1}^{N_e} \frac{Z_i e^2}{4\pi\varepsilon_0 \left|\vec{R}_i - \vec{r}_j\right|}}_{\text{electron-nuclei int.}} + \underbrace{\frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_e} \frac{e^2}{4\pi\varepsilon_0 \left|\vec{r}_i - \vec{r}_j\right|}}_{\text{electron-electron int.}} \tag{9}$$

- Hartree-Fock method allows for efficient mean-field treatment
- limited usefulness, does not incorporate electronic correlations
- can serve as a starting point for post-Hartree-Fock methods like configuration interaction or coupled cluster
- these methods are computationally hard due to the scaling w.r.t. system size
- only feasible for “small” molecules, typically up to $\mathcal{O}(10^2)$ electrons
- quantum computers can efficiently store and work with these correlated states

VQE: Unitary Coupled Cluster (UCC) Ansatz I

- wave function ansatz

$$|\Psi\rangle = e^{T-T^\dagger}\,|\Psi_{\mathrm{ref}}\rangle \tag{10}$$

- reference state $|\Psi_{\mathrm{ref}}\rangle$, typically a single Slater determinant of Hartree-Fock orbitals
- cluster operator

$$T = \sum_{i=1}^{N_e} T_i \tag{11}$$

- terms describe i-particle excitations to unoccupied (virtual) orbitals

$$T_1 = \sum_{\substack{i\in\mathrm{occ.}\\ a\in\mathrm{virt.}}} t_a^i a_a^\dagger a_i \tag{12}$$

$$T_2 = \sum_{\substack{i>j\in\mathrm{occ.}\\ a>b\in\mathrm{virt.}}} t_{ab}^{ij} a_a^\dagger a_b^\dagger a_i a_j \tag{13}$$

$$\dots \tag{14}$$

VQE: Unitary Coupled Cluster Ansatz (UCC) II

- the problem is to determine the coefficients t_a^i, t_{ab}^{ij}, ... that best describe the ground state (or any other desired state)

VQE: Unitary Coupled Cluster Ansatz (UCC) II

- the problem is to determine the coefficients t_a^i, t_{ab}^{ij}, ... that best describe the ground state (or any other desired state)
- usually one truncates T after some order of excitation $n < N_e$ (e.g. up to double excitations)
  - otherwise the problem might be too large
  - higher order terms contribute less
  - usually better truncation than configuration interaction (recovers more of the correlation energy due to the non-linear ansatz)

VQE: Unitary Coupled Cluster Ansatz (UCC) II

- the problem is to determine the coefficients t_a^i, t_{ab}^{ij}, ... that best describe the ground state (or any other desired state)
- usually one truncates T after some order of excitation $n < N_e$ (e.g. up to double excitations)
  - otherwise the problem might be too large
  - higher order terms contribute less
  - usually better truncation than configuration interaction (recovers more of the correlation energy due to the non-linear ansatz)
- the molecular Hamiltonian in second quantization reads

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqkl} h_{pqkl} a_p^\dagger a_q^\dagger a_k a_l \tag{15}$$

VQE: Unitary Coupled Cluster Ansatz (UCC) II

- the problem is to determine the coefficients t_a^i, t_{ab}^{ij}, ... that best describe the ground state (or any other desired state)
- usually one truncates T after some order of excitation $n < N_e$ (e.g. up to double excitations)
  - otherwise the problem might be too large
  - higher order terms contribute less
  - usually better truncation than configuration interaction (recovers more of the correlation energy due to the non-linear ansatz)
- the molecular Hamiltonian in second quantization reads

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqkl} h_{pqkl} a_p^\dagger a_q^\dagger a_k a_l \tag{15}$$

How to map the fermionic UCC function and the molecular Hamiltonian to a quantum computer (spin-½)?

VQE: Fermionic Transformations

- the fermionic operators $a_j^\dagger$ and a_j can be transformed to spin-$1/2$ operators using the Jordan-Wigner transformation

$$a_j^\dagger = \exp\left[i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^+ \tag{16}$$

$$a_j = \exp\left[-i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^- \tag{17}$$

with $\sigma_j^+ = 1/2(X_j + iY_j)$ and $\sigma_j^- = 1/2(X_j - iY_j)$

VQE: Fermionic Transformations

- the fermionic operators $a_j^\dagger$ and a_j can be transformed to spin-½ operators using the Jordan-Wigner transformation

$$a_j^\dagger = \exp\left[i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^+ \tag{16}$$

$$a_j = \exp\left[-i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^- \tag{17}$$

  with $\sigma_j^+ = 1/2(X_j + iY_j)$ and $\sigma_j^- = 1/2(X_j - iY_j)$
- alternative: Bravyi-Kitaev (BK) transformation

VQE: Fermionic Transformations

- the fermionic operators $a_j^\dagger$ and a_j can be transformed to spin-½ operators using the Jordan-Wigner transformation

$$a_j^\dagger = \exp\left[i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^+ \tag{16}$$

$$a_j = \exp\left[-i\pi \sum_{k=1}^{j-1} \sigma_k^+ \sigma_k^- \right] \sigma_j^- \tag{17}$$

  with $\sigma_j^+ = 1/2(X_j + iY_j)$ and $\sigma_j^- = 1/2(X_j - iY_j)$
- alternative: Bravyi-Kitaev (BK) transformation
- example: BK trafo for H_2

$$H = g_0 1 + g_1 Z_0 + g_2 Z_1 + g_3 Z_0 Z_1 + g_4 Y_0 Y_1 + g_5 X_0 X_1 \tag{18}$$

  (parameters g_i depend on nuclear coordinates and some constants)

VQE: Quantum Expectation Estimation I

How do we measure $\langle \Psi(\vec{\theta}) | H_{\mathrm{P}} | \Psi(\vec{\theta}) \rangle$?

- a quantum computer must be able to selectively measure Z_i
- other operators can be measured by performing unitary transformations U

$$O = U^{\dagger} Z U \tag{19}$$

- this allows for single-qubit Pauli measurements (measure X, Y, Z)

$$O = X \qquad U = H \tag{20}$$
$$O = Y \qquad U = HS^{\dagger} \tag{21}$$
$$O = Z \qquad U = 1 \tag{22}$$

here H denotes the Hadamard and S the $\pi/4$ phase gate

VQE: Quantum Expectation Estimation II

- the same principle can be applied to multi-qubit measurements

$$O = X \otimes Z \qquad U = \mathrm{CNOT}\,(H \otimes 1) \tag{23}$$

$$O = X \otimes Y \qquad U = \mathrm{CNOT}\left(H \otimes HS^{\dagger}\right) \tag{24}$$

$$\vdots \qquad\qquad \vdots$$

- this allows general Pauli strings to be measured, e.g. $X \otimes Y \otimes Y \otimes Z$

VQE: Quantum Expectation Estimation III

- let's assume the Hamiltonian consists of M Pauli strings

$$H = \sum_{\gamma=1}^{M} H_\gamma \tag{25}$$

- it's expectation value is given by

$$\langle H \rangle = \sum_{\gamma=1}^{M} \langle H_\gamma \rangle \tag{26}$$

- individual state preparation for every Pauli string $\Rightarrow$ additive variances

$$\mathrm{Var}[H] = \sum_{\gamma=1}^{M} \mathrm{Var}[H_\gamma] \tag{27}$$

VQE: Quantum Expectation Estimation IV

- assuming a normal distribution

$$n_{\text{preparations}} = M \sum_{\gamma=1}^{M} \frac{\mathsf{Var}[H_\gamma]}{\varepsilon^2} \tag{28}$$

  state preparations are required to achieve a variance smaller than ε
- usually the $\mathsf{Var}[H_\gamma]$ are unknown (but can be estimated)
- various strategies can be used to reduce the number of state preparations
- commutating terms can be measured in parallel
- for best results only terms with $\mathsf{Cov}[H_\alpha, H_\beta] = 0$ should be grouped, otherwise correlations might increase the number of necessary measurements

$$\text{Example}: \quad H = -X_0X_1 - Y_0Y_1 + Z_0Z_1 + Z_0 + Z_1 \tag{29}$$

$$\Rightarrow \{-X_0X_1\},\ \{-Y_0Y_1, Z_0Z_1\},\ \{Z_0, Z_1\} \tag{30}$$

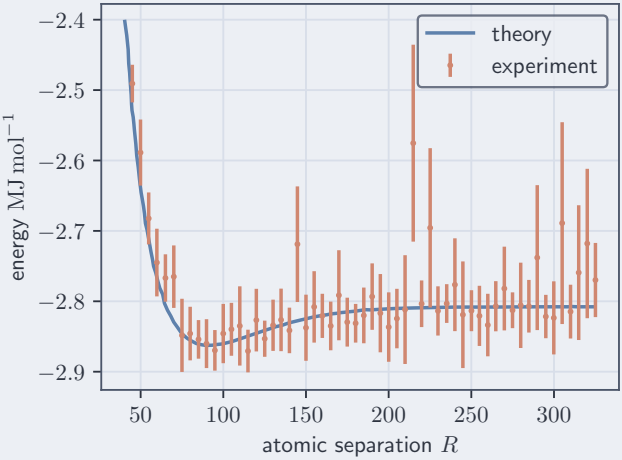
(covariance between X_0X_1 and Y_0Y_1)

VQE: Ingredients for Molecular Problem

We now have all ingredients for the electronic structure problem:

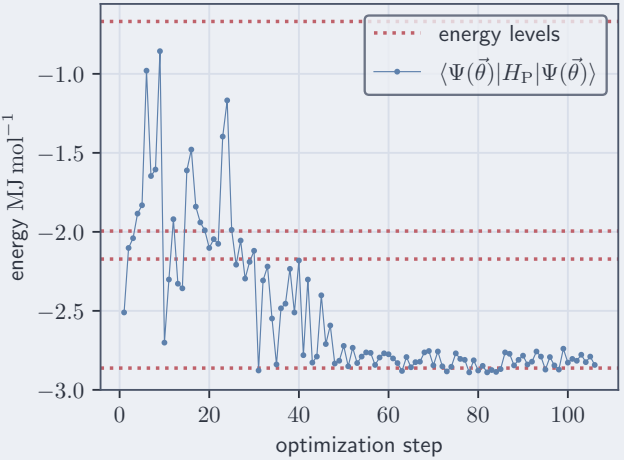
- trial state $|\Psi(\vec{\theta})\rangle$ is the UCC ansatz translated to Pauli matrices
- problem Hamiltonian H_{P} is the molecular Hamiltonian transformed to Pauli matrices
- optimization parameters are the UCC coefficients
- measurement of the observable can be achieved by clever measurement of Pauli strings

VQE: Example $\mathrm{He} - \mathrm{H}^+$



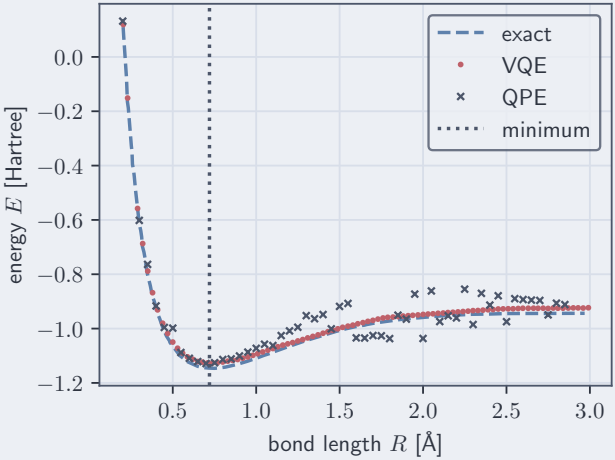
Data from Peruzzo et al. (2014), $\mathrm{He} - \mathrm{H}^+$ ground state energy using two photonic qubits

VQE: Example $\mathrm{He} - \mathrm{H}^+$ II



- convergence of the Nelder-Mead (simplex) algorithm
- gradient-free (evaluations of the cost function require costly state preparations)

VQE: Example H_2



Data from O'Malley et al. (2016), H_2 ground state energy using two superconducting qubits

Quantum Phase Estimation (QPE): Principle

Previous to the VQE ansatz QPE was used for electronic structure problems on quantum computers.

- given a unitary operator U, QPE estimates the phase ω in $U|\Psi\rangle = e^{2\pi i\omega}|\Psi\rangle$ of an eigenstate $|\Psi\rangle$

Quantum Phase Estimation (QPE): Principle

Previous to the VQE ansatz QPE was used for electronic structure problems on quantum computers.

- given a unitary operator U, QPE estimates the phase ω in $U|\Psi\rangle = e^{2\pi i\omega}|\Psi\rangle$ of an eigenstate $|\Psi\rangle$
- we can choose $U = \exp\left[{}^{iHt}/_{\hbar}\right]$, the phase ω determines the energy of the state

Quantum Phase Estimation (QPE): Principle

Previous to the VQE ansatz QPE was used for electronic structure problems on quantum computers.

- given a unitary operator U, QPE estimates the phase ω in $U|\Psi\rangle = e^{2\pi i\omega}|\Psi\rangle$ of an eigenstate $|\Psi\rangle$
- we can choose $U = \exp\left[{}^{iHt}/_{\hbar}\right]$, the phase ω determines the energy of the state
- the qubits of an additional register are iteratively entangled with the system register using controlled unitary gates
  - the system qubits act as control bits
  - the unitary is applied with different fixed phases 2^0, 2^1, 2^2, ...
  - the inverse quantum Fourier transformation is used to extract the phase from the ancillary register

Quantum Phase Estimation (QPE): Principle

Previous to the VQE ansatz QPE was used for electronic structure problems on quantum computers.

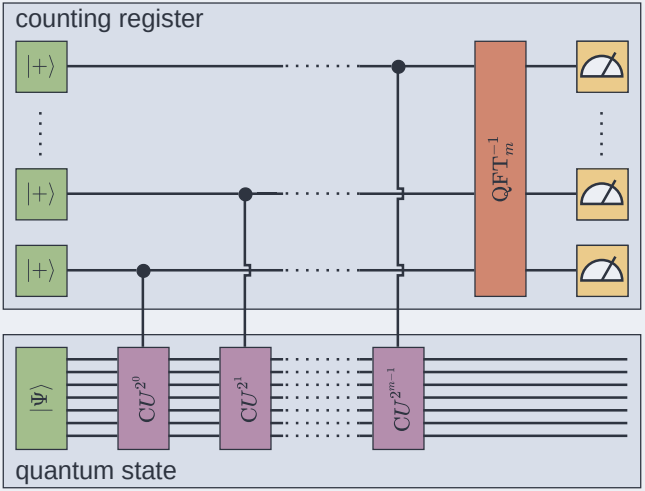
- given a unitary operator U, QPE estimates the phase ω in $U|\Psi\rangle = e^{2\pi i\omega}|\Psi\rangle$ of an eigenstate $|\Psi\rangle$
- we can choose $U = \exp\left[{}^{iHt}/_{\hbar}\right]$, the phase ω determines the energy of the state
- the qubits of an additional register are iteratively entangled with the system register using controlled unitary gates
  - the system qubits act as control bits
  - the unitary is applied with different fixed phases $2^0, 2^1, 2^2, \ldots$
  - the inverse quantum Fourier transformation is used to extract the phase from the ancillary register
- QPE reproduces ω with high probability within a margin ε using $\mathcal{O}\left(\log({}^{1}/_{\varepsilon})\right)$ ancillary qubits
- requires $\mathcal{O}({}^{1}/_{\varepsilon})$ controlled unitary gates

Quantum Phase Estimation (QPE): Principle

Previous to the VQE ansatz QPE was used for electronic structure problems on quantum computers.

- given a unitary operator U, QPE estimates the phase ω in $U|\Psi\rangle = e^{2\pi i\omega}|\Psi\rangle$ of an eigenstate $|\Psi\rangle$
- we can choose $U = \exp\left[{}^{iHt}/_{\hbar}\right]$, the phase ω determines the energy of the state
- the qubits of an additional register are iteratively entangled with the system register using controlled unitary gates
  - the system qubits act as control bits
  - the unitary is applied with different fixed phases $2^0, 2^1, 2^2, \ldots$
  - the inverse quantum Fourier transformation is used to extract the phase from the ancillary register
- QPE reproduces ω with high probability within a margin ε using $\mathcal{O}\left(\log({}^{1}/_{\varepsilon})\right)$ ancillary qubits
- requires $\mathcal{O}({}^{1}/_{\varepsilon})$ controlled unitary gates
- still requires a scheme to prepare $|\Psi\rangle$ close to the desired state

Quantum Phase Estimation (QPE): Sketch



Quantum Phase Estimation (QPE): Characteristics

Advantages:

- requires a single measurement
- in principle one could use QPE instead of QEE in VQE

Downsides:

- requires an extra qubit register
- the accuracy depends on the number of extra qubits m
- the number of required gates for the CU operations grows rapidly, quickly becomes infeasible due to limited decoherence times and gate fidelities
- CU gates are hard to implement
- if $2^m\omega$ is not an integer success probability is reduced from 1 to $\geq \frac{4}{\pi^2} \approx 0.405$

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- How to generate trial states?

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- objective function

$$C(\mathbf{z}) = \sum_{i=1}^{m} C_i(\mathbf{z}) \tag{32}$$

  given by m clauses $C_i(\mathbf{z})$

- How to generate trial states?

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- objective function

$$C(\mathbf{z}) = \sum_{i=1}^{m} C_i(\mathbf{z}) \tag{32}$$

  given by m clauses $C_i(\mathbf{z})$
- $C_i(\mathbf{z}) = 0$ if $\mathbf{z}$ does not fulfill the clause, 1 (or some finite value) otherwise

- How to generate trial states?

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- objective function

$$C(\mathbf{z}) = \sum_{i=1}^{m} C_i(\mathbf{z}) \tag{32}$$

  given by m clauses $C_i(\mathbf{z})$
- $C_i(\mathbf{z}) = 0$ if $\mathbf{z}$ does not fulfill the clause, 1 (or some finite value) otherwise
- goal: find the $\mathbf{z}$ that fulfills the maximum number of clauses (MAX-SAT problem)

- How to generate trial states?

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- objective function

$$C(\mathbf{z}) = \sum_{i=1}^{m} C_i(\mathbf{z}) \tag{32}$$

  given by m clauses $C_i(\mathbf{z})$
- $C_i(\mathbf{z}) = 0$ if $\mathbf{z}$ does not fulfill the clause, 1 (or some finite value) otherwise
- goal: find the $\mathbf{z}$ that fulfills the maximum number of clauses (MAX-SAT problem)
- each clause typically depends only on a few bits

$$C_1(\mathbf{z}) = z_1 \vee z_2, \quad C_2(\mathbf{z}) = z_2 \vee \bar{z}_3, \quad C_3(\mathbf{z}) = z_1 \vee \bar{z}_3 \vee z_4, \quad \dots \tag{33}$$

VQE: Classical Binary Optimization Problems

- optimization problem whose solutions are bit strings (N bits)

$$\mathbf{z} = z_1 z_2 \dots z_n \quad z_i \in \{0, 1\} \tag{31}$$

- objective function

$$C(\mathbf{z}) = \sum_{i=1}^{m} C_i(\mathbf{z}) \tag{32}$$

  given by m clauses $C_i(\mathbf{z})$
- $C_i(\mathbf{z}) = 0$ if $\mathbf{z}$ does not fulfill the clause, 1 (or some finite value) otherwise
- goal: find the $\mathbf{z}$ that fulfills the maximum number of clauses (MAX-SAT problem)
- each clause typically depends only on a few bits
- $z_i \to \frac{1}{2}(1 - Z_i)$ maps cost function to an Ising Hamiltonian, can be treated with VQE ansatz

$$H_{\mathrm{P}} = \sum_{i=1}^{N} h_i Z_i + \sum_{i,j=1}^{N} h_{ij} Z_i Z_j + \sum_{ijk=1}^{N} h_{ijk} Z_i Z_j Z_k + \dots \tag{33}$$

- How to generate trial states?

Quantum Approximate Optimization Algorithm (QAOA)

QAOA: Approximation Algorithms

- many classical binary optimization problems are computationally hard
- consider approximation algorithms that yield non-optimal but good solutions

QAOA: Approximation Algorithms

- many classical binary optimization problems are computationally hard
- consider approximation algorithms that yield non-optimal but good solutions
- let $C_{\max} = \max_{z} C(z)$ be the optimal solution

QAOA: Approximation Algorithms

- many classical binary optimization problems are computationally hard
- consider approximation algorithms that yield non-optimal but good solutions
- let $C_{\max} = \max_{\mathbf{z}} C(\mathbf{z})$ be the optimal solution
- the quality of a proposed solution $\mathbf{z}$ is given by the approximation ratio

$$r = \frac{C(\mathbf{z})}{C_{\max}} \tag{34}$$

- $r = 1$ corresponds to optimality

QAOA: Approximation Algorithms

- many classical binary optimization problems are computationally hard
- consider approximation algorithms that yield non-optimal but good solutions
- let $C_{\max} = \max_{\mathbf{z}} C(\mathbf{z})$ be the optimal solution
- the quality of a proposed solution $\mathbf{z}$ is given by the approximation ratio

$$r = \frac{C(\mathbf{z})}{C_{\max}} \tag{34}$$

- $r = 1$ corresponds to optimality
- if $r \geq r^*$ for all problem instances: r^* characterizes the quality of the approximation algorithm

QAOA: Ansatz

QAOA offers a heuristic ansatz to solve classical binary optimization problems encoded as Ising Hamiltonians

$$H_{\mathrm{P}} = \sum_{i=1}^{N} h_i Z_i + \sum_{i,j=1}^{N} h_{ij} Z_i Z_j + \sum_{ijk=1}^{N} h_{ijk} Z_i Z_j Z_k + \dots \tag{35}$$

and employs a mixing Hamiltonian

$$H_{\mathrm{M}} = \sum_{i=1}^{N} X_i. = \bigotimes_{i=1}^{N} e^{-i\beta_j X_i} = \bigotimes_{i=1}^{N} R_x(2\beta_i) \tag{36}$$

QAOA: Ansatz

QAOA offers a heuristic ansatz to solve classical binary optimization problems encoded as Ising Hamiltonians

$$H_{\mathrm{P}} = \sum_{i=1}^{N} h_i Z_i + \sum_{i,j=1}^{N} h_{ij} Z_i Z_j + \sum_{ijk=1}^{N} h_{ijk} Z_i Z_j Z_k + \dots \tag{35}$$

and employs a mixing Hamiltonian

$$H_{\mathrm{M}} = \sum_{i=1}^{N} X_i . = \bigotimes_{i=1}^{N} e^{-i\beta_j X_i} = \bigotimes_{i=1}^{N} R_{\mathrm{x}}(2\beta_i) \tag{36}$$

A single integer parameter p (circuit depth) controls the trial state ansatz

$$|\Psi(\vec{\beta}, \vec{\gamma})\rangle = e^{-i\beta_p H_{\mathrm{M}}} e^{-i\gamma_p H_{\mathrm{P}}} \dots e^{-i\beta_1 H_{\mathrm{M}}} e^{-i\gamma_1 H_{\mathrm{P}}} |\Psi_0\rangle \tag{37}$$

that is determined by $2p$ parameters $0 \leq \gamma_i < 2\pi$ and $0 \leq \beta_i \leq \pi$.

QAOA: Ansatz

QAOA offers a heuristic ansatz to solve classical binary optimization problems encoded as Ising Hamiltonians

$$H_{\mathrm{P}} = \sum_{i=1}^{N} h_i Z_i + \sum_{i,j=1}^{N} h_{ij} Z_i Z_j + \sum_{ijk=1}^{N} h_{ijk} Z_i Z_j Z_k + \dots \tag{35}$$

and employs a mixing Hamiltonian

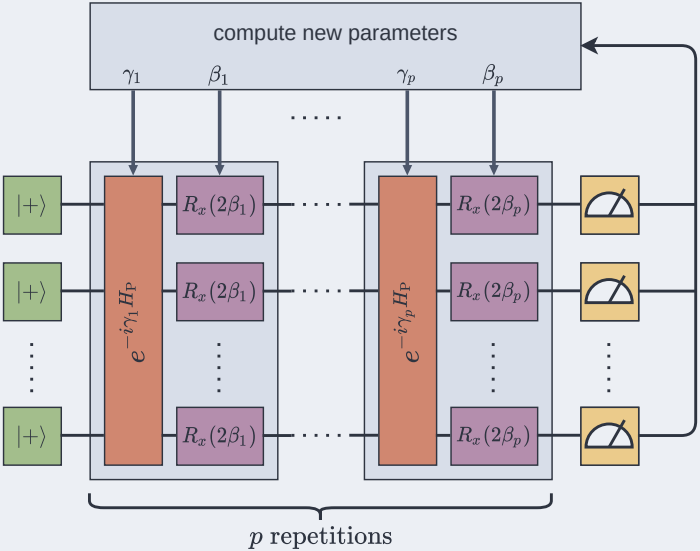
$$H_{\mathrm{M}} = \sum_{i=1}^{N} X_i. \qquad e^{-i\beta_j H_{\mathrm{M}}} = \bigotimes_{i=1}^{N} e^{-i\beta_j X_i} = \bigotimes_{i=1}^{N} R_{\mathrm{x}}(2\beta_i) \tag{36}$$

A single integer parameter p (circuit depth) controls the trial state ansatz

$$|\Psi(\vec{\beta}, \vec{\gamma})\rangle = e^{-i\beta_p H_{\mathrm{M}}} e^{-i\gamma_p H_{\mathrm{P}}} \dots e^{-i\beta_1 H_{\mathrm{M}}} e^{-i\gamma_1 H_{\mathrm{P}}} |\Psi_0\rangle \tag{37}$$

that is determined by $2p$ parameters $0 \leq \gamma_i < 2\pi$ and $0 \leq \beta_i \leq \pi$.

QAOA: Circuit Visualization



QAOA: Optimization Problem

- the parameters $\vec{\beta}$ and $\vec{\gamma}$ are optimized to maximize the expectation value

$$F_p = \langle \Psi(\vec{\beta}, \vec{\gamma}) | H_{\mathrm{P}} | \Psi(\vec{\beta}, \vec{\gamma}) \rangle \tag{38}$$

- approximation ratio

$$r_p = \frac{F_p}{C_{\max}} \tag{39}$$

- adding a layer cannot worsen the result

$$F_{p+1} \geq F_p \tag{40}$$

- optimal limit

$$\lim_{p \to \infty} F_p = C_{\max} \tag{41}$$

  approximates adiabatic quantum computing

QAOA: Motivation I

- recall adiabatic quantum computing:

$$H(t) = \big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}} \tag{42}$$

QAOA: Motivation I

- recall adiabatic quantum computing:

$$H(t) = \big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}} \tag{42}$$

- discrete time-step

$$|\Psi(t + \Delta t)\rangle = \exp\Big[-i\Delta t \left(\big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}}\right)\Big] |\Psi(t)\rangle \,. \tag{43}$$

QAOA: Motivation I

- recall adiabatic quantum computing:

$$H(t) = \big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}} \tag{42}$$

- discrete time-step

$$|\Psi(t + \Delta t)\rangle = \exp\Big[-i\Delta t \left(\big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}}\right)\Big] |\Psi(t)\rangle \,. \tag{43}$$

- Trotter-Suzuki first order

$$e^{\delta(A+B)} \approx e^{\delta A} e^{\delta B} + \mathcal{O}(\delta^2) \tag{44}$$

QAOA: Motivation I

- recall adiabatic quantum computing:

$$H(t) = \big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}} \tag{42}$$

- discrete time-step

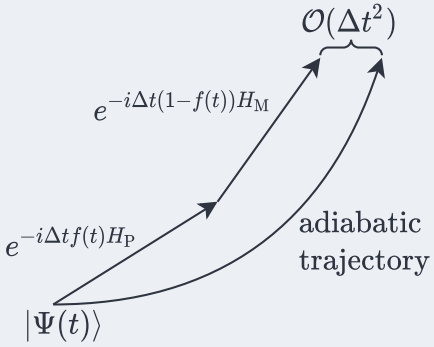
$$|\Psi(t + \Delta t)\rangle = \exp\Big[-i\Delta t \left(\big(1 - f(t)\big) H_{\mathrm{M}} + f(t) H_{\mathrm{P}}\right)\Big] |\Psi(t)\rangle \,. \tag{43}$$

- Trotter-Suzuki first order

$$e^{\delta(A+B)} \approx e^{\delta A} e^{\delta B} + \mathcal{O}(\delta^2) \tag{44}$$

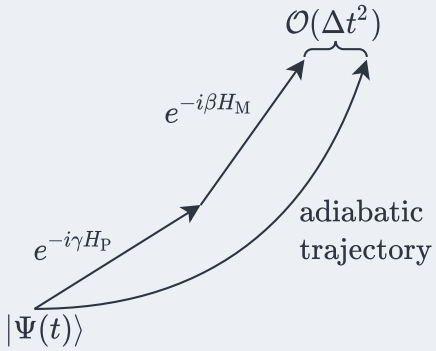
$$|\Psi(t + \Delta t)\rangle \approx \exp\Big[-i\big(1 - f(t)\big)\Delta t H_{\mathrm{M}}\Big] \exp\Big[-if(t)\Delta t H_{\mathrm{P}}\Big] |\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{45}$$

$$|\Psi(t+\Delta t)\rangle \approx \exp\Big[-i\big(1-f(t)\big)\Delta t H_{\mathrm{M}}\Big]\exp\Big[-if(t)\Delta t H_{\mathrm{P}}\Big]\,|\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{46}$$



$$|\Psi(t+\Delta t)\rangle \approx \exp\left[-i\beta H_{\mathrm{M}}\right] \exp\left[-i\gamma H_{\mathrm{P}}\right] |\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{46}$$

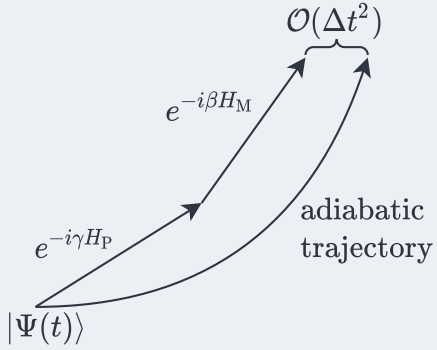
- define $\beta = (1 - f(t))\,\Delta t$ and $\gamma = f(t)\Delta t$



QAOA: Motivation II

$$|\Psi(t+\Delta t)\rangle \approx \underbrace{\exp\left[-i\beta H_{\mathrm{M}}\right] \exp\left[-i\gamma H_{\mathrm{P}}\right]}_{\text{one QAOA layer}} |\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{46}$$

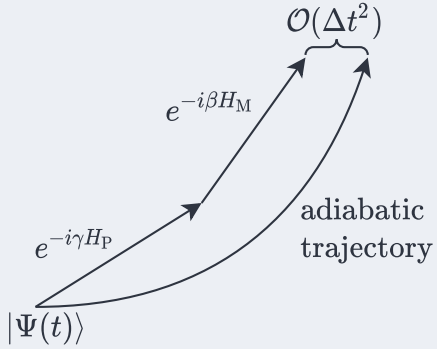
- define $\beta = (1 - f(t))\,\Delta t$ and $\gamma = f(t)\Delta t$
- one time-step $\widehat{=}$ one layer of QAOA



QAOA: Motivation II

$$|\Psi(t+\Delta t)\rangle \approx \underbrace{\exp\left[-i\beta H_{\mathrm{M}}\right] \exp\left[-i\gamma H_{\mathrm{P}}\right]}_{\text{one QAOA layer}} |\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{46}$$

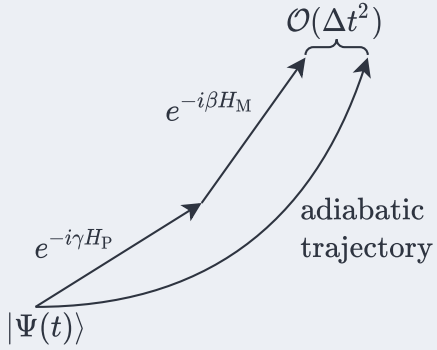
- define $\beta = (1 - f(t))\,\Delta t$ and $\gamma = f(t)\Delta t$
- one time-step $\widehat{=}$ one layer of QAOA
- however QAOA optimizes β and γ independently



QAOA: Motivation II

$$|\Psi(t+\Delta t)\rangle \approx \underbrace{\exp\left[-i\beta H_{\mathrm{M}}\right] \exp\left[-i\gamma H_{\mathrm{P}}\right]}_{\text{one QAOA layer}} |\Psi(t)\rangle + \mathcal{O}(\Delta t^2) \tag{46}$$

- define $\beta = (1 - f(t))\,\Delta t$ and $\gamma = f(t)\Delta t$
- one time-step $\widehat{=}$ one layer of QAOA
- however QAOA optimizes β and γ independently
- $p \to \infty$ (infinitely many layers) reproduces adiabatic trajectory



QAOA: Applications

- maximum likelihood channel decoding (communications)
- community detection (social, neural and biological networks)
- portfolio optimization (finance industry)
- exact cover problem (tiling problems and aircraft flight gate assignment)
- maximum independent set (radio networks and genetic engineering)
- MaxCut (integrated circuit design, statistical physics, data clustering)
- linear algebra (fundamental for many applications)
- traveling salesperson (logistics, traveling)
- graph coloring (scheduling, compiler optimization)
- correlation clustering (data science)
- other satisfiability problems, ...

QAOA: Advantages

- well controllable
  - single integer parameter determines the circuit structure
  - increasing p can not worsen solution (monotonous)
- it has been shown that the parametric square-pulse ansatz (“bang-bang”) is optimal given a fixed quantum computation time budget
- QAOA can learn to exploit diabatic transitions to overcome small energy gap limitations
- gate imperfections can be mitigated by parameter tuning
- circuits are relatively shallow
- nice implementation on Rydberg platform (driving with global pulses)

Max-Cut

Max-Cut: Problem Statement

Partition the vertices of a graph in two sets, such that the sum of edge weights between the sets becomes maximal.

$$C(\mathbf{z}) = \sum_{i,j=1}^{N} w_{ij} z_i (1 - z_j) \tag{47}$$

Max-Cut: Problem Statement

Partition the vertices of a graph in two sets, such that the sum of edge weights between the sets becomes maximal.

$$C(\mathbf{z}) = \sum_{i,j=1}^{N} w_{ij} z_i (1 - z_j) \tag{47}$$

- NP-hard: no polynomial time algorithm (if N $\neq$ NP)
- state-of-the-art solvers can solve up to $N \approx 100$
- stochastic solvers for $N = \mathcal{O}(10^2)$ (no solution guaranteed)

Max-Cut: Problem Statement

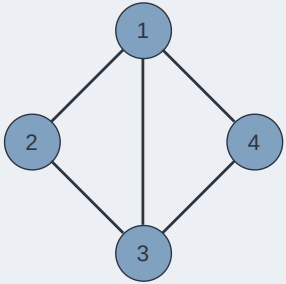
Partition the vertices of a graph in two sets, such that the sum of edge weights between the sets becomes maximal.

$$C(\mathbf{z}) = \sum_{i,j=1}^{N} w_{ij} z_i (1 - z_j) \tag{47}$$

- NP-hard: no polynomial time algorithm (if N $\neq$ NP)
- state-of-the-art solvers can solve up to $N \approx 100$
- stochastic solvers for $N = \mathcal{O}(10^2)$ (no solution guaranteed)
- APX-hard: every polynomial-time approximation scheme has an approximation ratio guarantee $r^* < 1$
- solving the problem for generic graphs with $r^* \geq {}^{16}/_{17} \approx 0.9412$ is NP-hard
- the best classical algorithm can only guarantee $r^* \gtrapprox 0.87856$

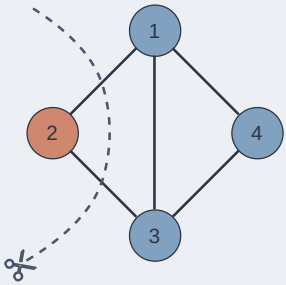
Max-Cut: Example

Here: equal weights $w_{ij} = 1$



Max-Cut: Example

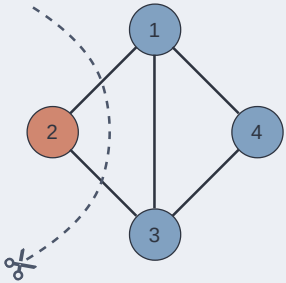
Here: equal weights $w_{ij} = 1$



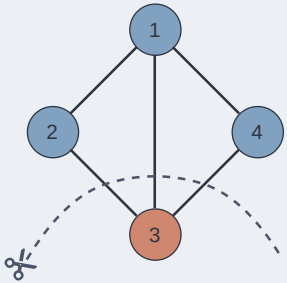
two edges

Max-Cut: Example

Here: equal weights $w_{ij} = 1$



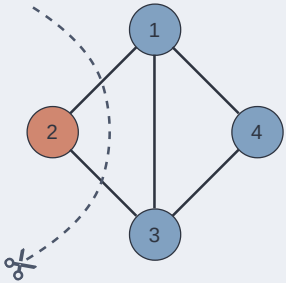
two edges



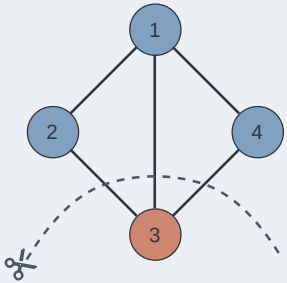
three edges

Max-Cut: Example

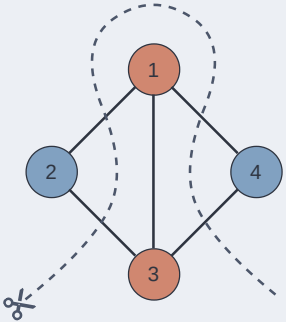
Here: equal weights $w_{ij} = 1$



two edges



three edges



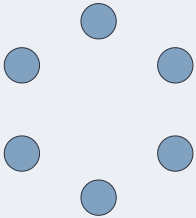
four edges

Max-Cut: k-Regular Graphs

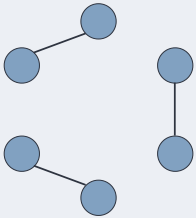
- often: only consider k-regular graphs (usually $k = 3$), i.e. graphs where each vertex has exactly 3 edges to other vertices.

Max-Cut: k-Regular Graphs

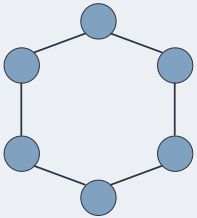
- often: only consider k-regular graphs (usually $k = 3$), i.e. graphs where each vertex has exactly 3 edges to other vertices.



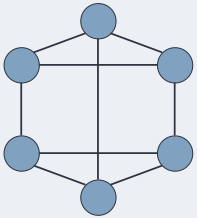
$k = 0$



$k = 1$



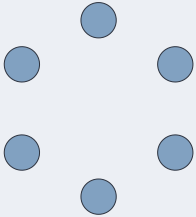
$k = 2$



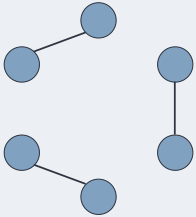
$k = 3$

Max-Cut: k-Regular Graphs

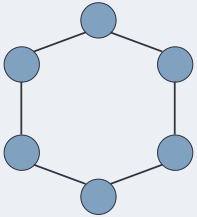
- often: only consider k-regular graphs (usually $k = 3$), i.e. graphs where each vertex has exactly 3 edges to other vertices.
- solving with $r^* \geq {}^{331}/_{332} \approx 0.9970$ is NP-hard



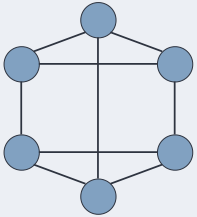
$k = 0$



$k = 1$



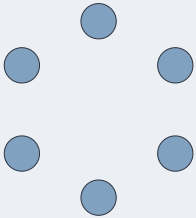
$k = 2$



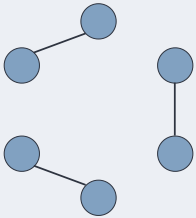
$k = 3$

Max-Cut: k-Regular Graphs

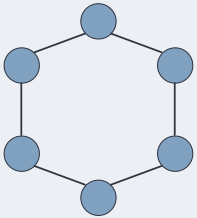
- often: only consider k-regular graphs (usually $k = 3$), i.e. graphs where each vertex has exactly 3 edges to other vertices.
- solving with $r^* \geq {}^{331}/_{332} \approx 0.9970$ is NP-hard
- best classical algorithm guarantees $r \gtrapprox 0.9326$



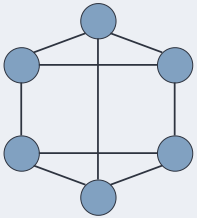
$k = 0$



$k = 1$



$k = 2$



$k = 3$

Max-Cut: Application of QAOA

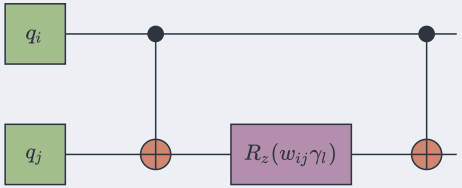
- many studies investigating this problem (including original QAOA paper)
- the problem can be encoded as the Ising Hamiltonian

$$H_{\mathrm{P}} = \frac{1}{2} \sum_{\langle i,j \rangle} w_{ij} \left(1 - Z_i Z_j\right) \tag{48}$$

- problem instance specific constant

$$C = \frac{1}{2} \sum_{\langle i,j \rangle} w_{ij} \tag{49}$$

- sub-circuit for each edge of the graph ($w_{ij} = 1$): $\exp\left[-\frac{i}{2} w_{ij} \gamma_l Z_i Z_j\right]$



Max-Cut: QAOA Results

Performance guarantees for 3-regular graphs (lower bounds):

- Farhi et al. (2014): $r^* \geq 0.692$ for $p = 1$

Max-Cut: QAOA Results

Performance guarantees for 3-regular graphs (lower bounds):

- Farhi et al. (2014): $r^* \geq 0.692$ for $p = 1$
- Wurtz & Love (2021):
  - $r^* > 0.7559$ for $p = 2$
  - $r^* > 0.7924$ for $p = 3$

Max-Cut: QAOA Results

Performance guarantees for 3-regular graphs (lower bounds):

- Farhi et al. (2014): $r^* \geq 0.692$ for $p = 1$
- Wurtz & Love (2021):
  - $r^* > 0.7559$ for $p = 2$
  - $r^* > 0.7924$ for $p = 3$
  - no quantum advantage for at least $p < 6$

Max-Cut: QAOA Results

Performance guarantees for 3-regular graphs (lower bounds):

- Farhi et al. (2014): $r^* \geq 0.692$ for $p = 1$
- Wurtz & Love (2021):
  - $r^* > 0.7559$ for $p = 2$
  - $r^* > 0.7924$ for $p = 3$
  - no quantum advantage for at least $p < 6$
- no known bounds for $p > 3$

Max-Cut: QAOA Results

Performance guarantees for 3-regular graphs (lower bounds):

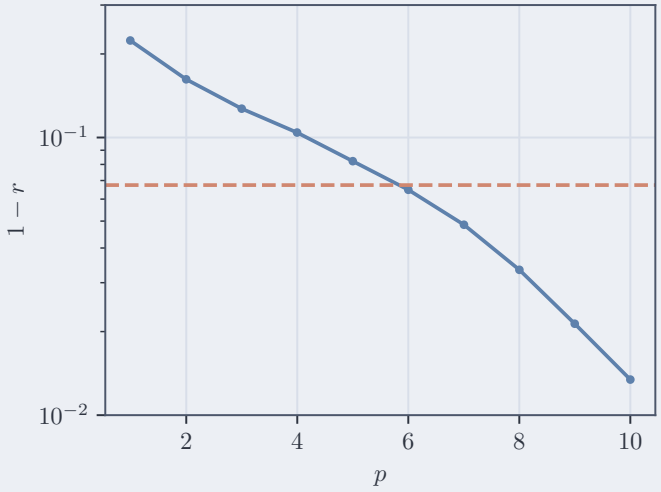
- Farhi et al. (2014): $r^* \geq 0.692$ for $p = 1$
- Wurtz & Love (2021):
  - $r^* > 0.7559$ for $p = 2$
  - $r^* > 0.7924$ for $p = 3$
  - no quantum advantage for at least $p < 6$
- no known bounds for $p > 3$

Zhou et al. (2020):

- robustness against small energy gaps
- exploit patterns in parameters

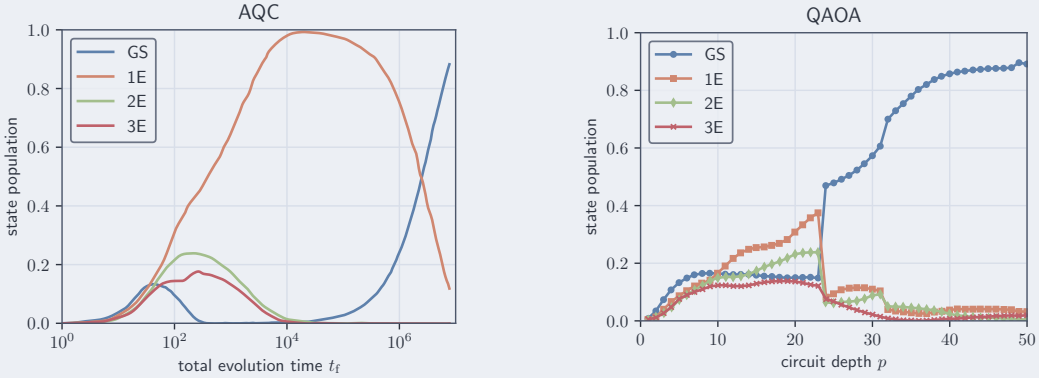
Max-Cut: Approximation Ratio for different p

Performance on weighted 4-regular graphs with 16 vertices.



Max-Cut: Small Energy Gaps

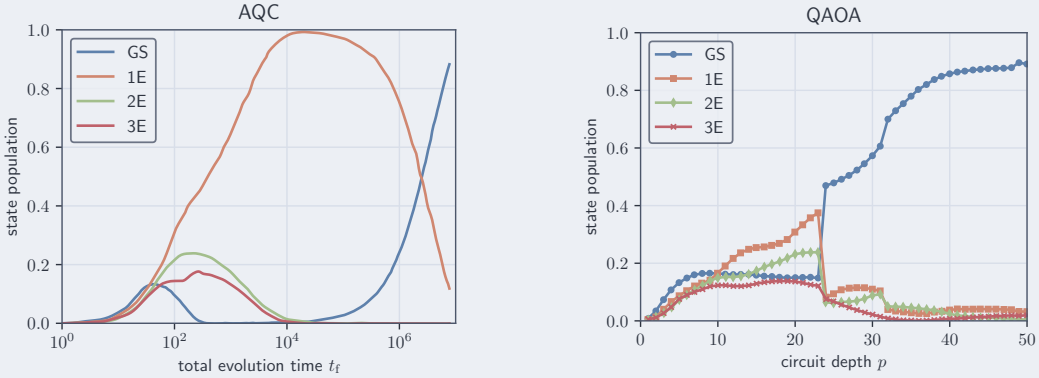
AQC requires extremely long evolution times when using a simple trajectory



Data from Zhou et al. (2020), 3-regular graph with $N = 14$ nodes and a small minimum spectral gap $\Delta E < 10^{-3}$

Max-Cut: Small Energy Gaps

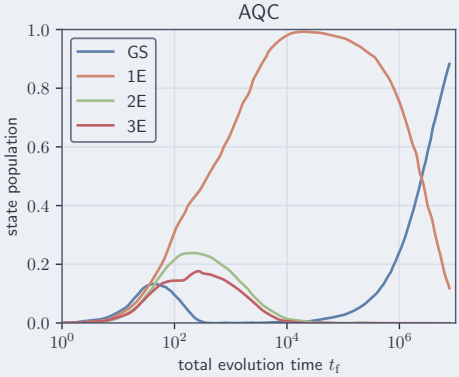
QAOA is reasonably robust against the small energy gap

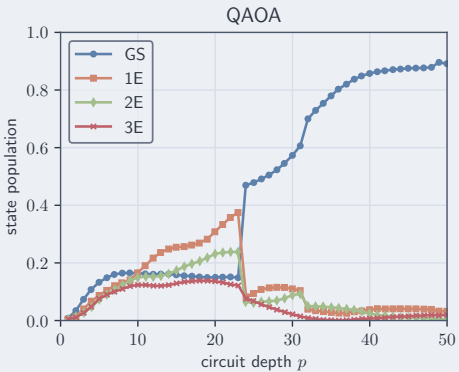


Data from Zhou et al. (2020), 3-regular graph with $N = 14$ nodes and a small minimum spectral gap $\Delta E < 10^{-3}$

Max-Cut: Small Energy Gaps

$p \gtrsim 24$ significantly improves the result

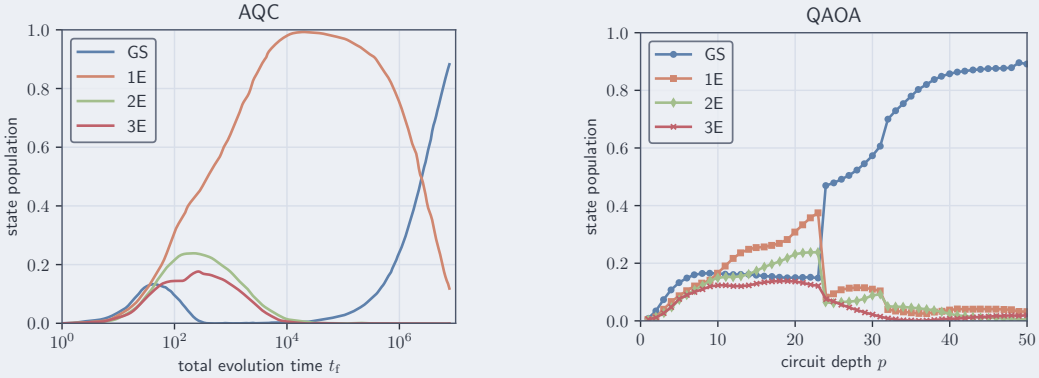




Data from Zhou et al. (2020), 3-regular graph with $N = 14$ nodes and a small minimum spectral gap $\Delta E < 10^{-3}$

Max-Cut: Small Energy Gaps

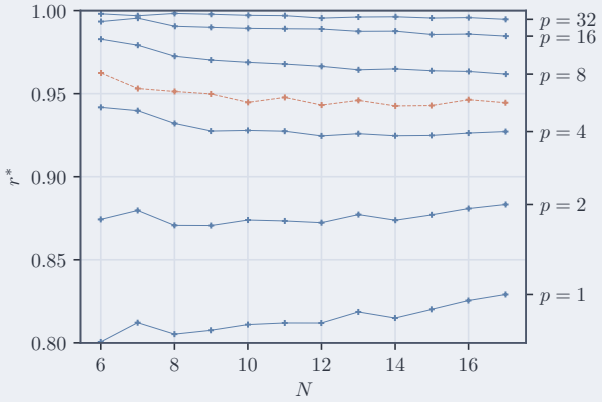
QAOA result can be used to craft an optimized trajectory for AQC



Data from Zhou et al. (2020), 3-regular graph with $N = 14$ nodes and a small minimum spectral gap $\Delta E < 10^{-3}$

Max-Cut: Random Graphs (Erdős–Rényi)

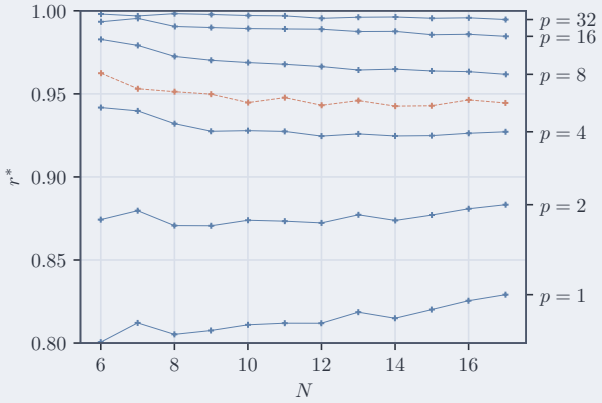
QAOA can also perform well on random graphs



orange: best classical algorithm, data from Crooks (2018)

Max-Cut: Random Graphs (Erdős–Rényi)

Here $p \geq 8$ achieves quantum advantage (**note system size however!**)



orange: best classical algorithm, data from Crooks (2018)

Conclusions & Outlook

Conclusions & Outlook: VQE

Conclusions:

- VQE has been used to accurately reproduce the energies of simple molecules
- with current hardware it produces more accurate results than QPE

Conclusions & Outlook: VQE

Conclusions:

- VQE has been used to accurately reproduce the energies of simple molecules
- with current hardware it produces more accurate results than QPE

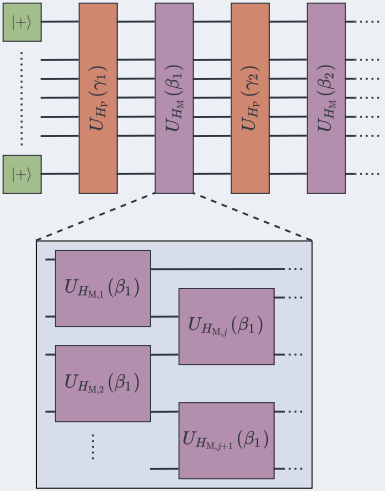
Outlook:

- treatment of larger molecules:
  - largest so far 6 qubits for BeH_2 (heuristic ansatz instead of UCC)
  - far from any actual quantum advantage
- deeper analysis of convergence and local optima

Outlook: QAOA

Outlook:

- better understanding for $p > 1$
- optimizing mixer Hamiltonian
- improved strategies for optimizing $\vec{\gamma}/\vec{\beta}$
- demonstrate quantum advantage
  - surpass classical approximation ratio on system sizes that cannot be treated with exact algorithms
  - actual experimental implementations
  - “soon”: QAOA for $N = 400$ vertices with $p \gtrapprox 25$ (Rydberg atoms in optical tweezer arrays)



Select References: VQE

- Original work, $\mathrm{He} - \mathrm{H}^+$ using two photonic qubits
  Alberto Peruzzo et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor”. en. In: *Nature Communications* 5.1 (July 2014), p. 4213. ISSN: 2041-1723. DOI: 10/f6df9g
- H_2 using two superconducting qubits
  P. J. J. O’Malley et al. “Scalable Quantum Simulation of Molecular Energies”. In: *Phys. Rev. X* 6.3 (July 2016), p. 031007. DOI: 10/f8wcgg
- theoretical discussion of the method
  Jarrod R. McClean et al. “The Theory of Variational Hybrid Quantum-Classical Algorithms”. en. In: *New J. Phys.* 18.2 (Feb. 2016), p. 023023. ISSN: 1367-2630. DOI: 10/gdcc5s

Select References: QAOA

- Original work, Max-Cut with QAOA
  Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A Quantum Approximate Optimization Algorithm”. In: *arXiv:1411.4028 [quant-ph]* (Nov. 2014). arXiv: 1411.4028 [quant-ph]
- In-depth analysis, comparison to AQC
  Leo Zhou et al. “Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices”. In: *Phys. Rev. X* 10.2 (June 2020), p. 021067. DOI: 10/gg4nk2
- Random graphs
  Gavin E. Crooks. “Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem”. In: *arXiv:1811.08419 [quant-ph]* (Nov. 2018). arXiv: 1811.08419 [quant-ph]

Thank you for your attention!