Introduction to Quantum Computing

Lecture 2

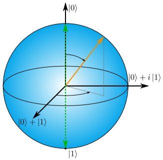
Stefan Kühn

DESY Summer Student Program, 01.08.2023

Recap of Lecture I

Qubits

- Quantum mechanical two level systems $\mathcal{H} = \{|0\rangle, |1\rangle\}$
- Can be in superposition
- Qubits can be entangled



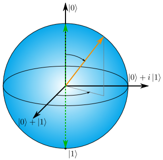
Recap of Lecture I

Qubits

> Quantum mechanical two level systems $\mathcal{H} = \{|0\rangle, |1\rangle\}$

> Can be in superposition

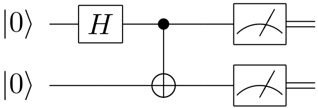
> Qubits can be entangled



Quantum gates and quantum circuits

> Quantum gates: unitary operations on a single/few qubits

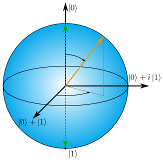
> Combining quantum gates we can express any unitary operation



Recap of Lecture I

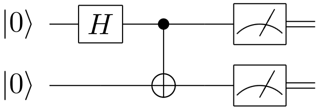
Qubits

- Quantum mechanical two level systems $\mathcal{H} = \{|0\rangle, |1\rangle\}$
- Can be in superposition
- Qubits can be entangled



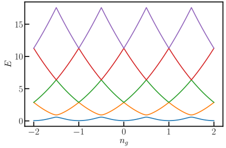
Quantum gates and quantum circuits

- Quantum gates: unitary operations on a single/few qubits
- Combining quantum gates we can express any unitary operation



Realization of qubits in physical systems

- Neutral atoms
- Trapped ions
- Superconducting qubits



Outline

Superdense coding

The Deutsch-Josza algorithm

Complexity theory

Hybrid quantum-classical algorithms

Superdense coding

Setting

- Alice wants to communicate two classical bits to Bob


- Classically we have to send (at least) two bits
- Can we do better using quantum bits?

Superdense coding

Bell states

- There are four Bell states

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle\big), \qquad \left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle\big)$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle\big), \qquad \left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle\big)$$

- All Bell states represent a maximally entangled pair of qubits
- They form a basis for the Hilbert space of two qubits

Superdense coding

Bell states

- There are four Bell states

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle\big), \qquad \left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle\big)$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle\big), \qquad \left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle\big)$$

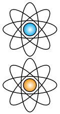
- All Bell states represent a maximally entangled pair of qubits
- They form a basis for the Hilbert space of two qubits
- Applying one of the Pauli gates X, Y or Z to one of the qubits transforms them into each other (up to global phases)

$$X \otimes \mathbb{1} \left|\Phi^+\right\rangle = \left|\Psi^+\right\rangle, \qquad Y \otimes \mathbb{1} \left|\Phi^+\right\rangle = -i \left|\Psi^-\right\rangle, \qquad Z \otimes \mathbb{1} \left|\Phi^+\right\rangle = \left|\Phi^-\right\rangle$$

Superdense coding

Superdense coding

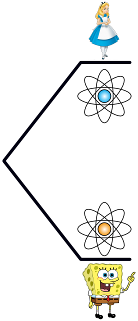
- Prepare a Bell state, e.g. $|\Phi^+\rangle$



Superdense coding

Superdense coding

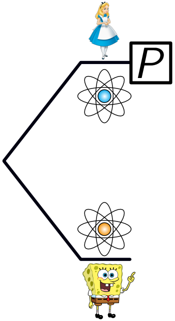
- Prepare a Bell state, e.g. $|\Phi^+\rangle$
- Alice and Bob each get one qubit of the Bell state



Superdense coding

Superdense coding

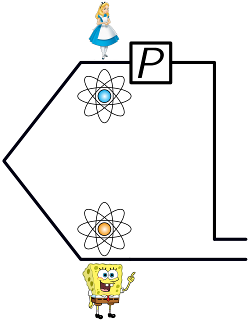
- Prepare a Bell state, e.g. $|\Phi^+\rangle$
- Alice and Bob each get one qubit of the Bell state
- Alice applies a Pauli gate on her qubit depending on the classical bitstring she wants to send
  - $\mathbb{1}$ for 00
  - X for 01
  - Y for 10
  - Z for 11



Superdense coding

Superdense coding

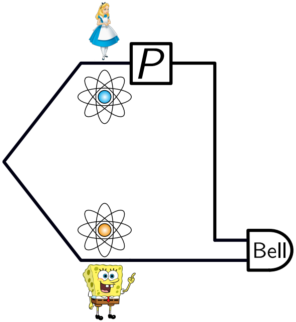
- Prepare a Bell state, e.g. $|\Phi^+\rangle$
- Alice and Bob each get one qubit of the Bell state
- Alice applies a Pauli gate on her qubit depending on the classical bitstring she wants to send
  - $\mathbb{1}$ for 00
  - X for 01
  - Y for 10
  - Z for 11
- Alice sends her qubit to Bob



Superdense coding

Superdense coding

- Prepare a Bell state, e.g. $|\Phi^+\rangle$
- Alice and Bob each get one qubit of the Bell state
- Alice applies a Pauli gate on her qubit depending on the classical bitstring she wants to send
  - $\mathbb{1}$ for 00
  - X for 01
  - Y for 10
  - Z for 11
- Alice sends her qubit to Bob
- Bob performs a Bell measurement which tells him Alice's bit string

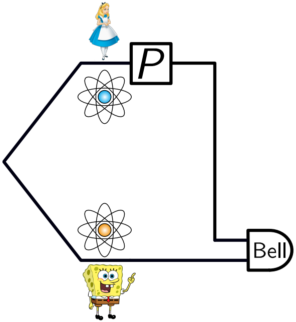


Superdense coding

Superdense coding

- Prepare a Bell state, e.g. $|\Phi^+\rangle$
- Alice and Bob each get one qubit of the Bell state
- Alice applies a Pauli gate on her qubit depending on the classical bitstring she wants to send
  - $\mathbb{1}$ for 00
  - X for 01
  - Y for 10
  - Z for 11
- Alice sends her qubit to Bob
- Bob performs a Bell measurement which tells him Alice's bit string

$\Rightarrow$ We only need a single qubit!



Superdense coding

Superdense coding

- The entangled Bell pair is a resource
- It can be shared long before the communication should take place

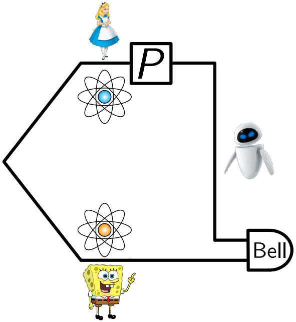
Superdense coding

Superdense coding

- The entangled Bell pair is a resource
- It can be shared long before the communication should take place
- Communication is secure
  - If an eavesdropper gets access to Alice's qubit they only have one of the qubits
  - As we have seen the reduced density operator for a single qubit of Bell state is maximally mixed

$$\rho_1 = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right)$$

⇒ The eavesdropper cannot infer information about the classical bit string



2.

Superdense coding

The Deutsch-Josza algorithm

Complexity theory

Hybrid quantum-classical algorithms

The Deutsch-Josza algorithm

Setting

- Given: a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ that is promised to be constant or balanced
- Task: find out if f is constant or balanced

The Deutsch-Josza algorithm

Setting

- Given: a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ that is promised to be constant or balanced
- Task: find out if f is constant or balanced

- Classical computer: try more than half of the possible inputs
  $\Rightarrow \frac{1}{2} \times 2^n + 1 = 2^{n-1} + 1$ function calls

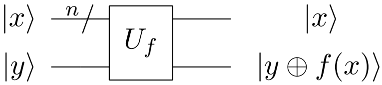
x_0	x_1	$f(x_0, x_1)$
0	0	1
0	1	1
1	0	0
1	1	0

The Deutsch-Josza algorithm

Setting

- Given: a function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ that is promised to be constant or balanced
- Task: find out if f is constant or balanced

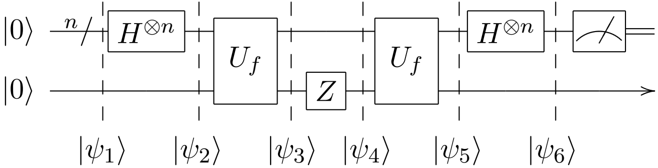
- Classical computer: try more than half of the possible inputs
  $\Rightarrow \frac{1}{2} \times 2^n + 1 = 2^{n-1} + 1$ function calls
- Let us assume we have a unitary $U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle$



- U_f is called an **oracle**

The Deutsch-Josza algorithm

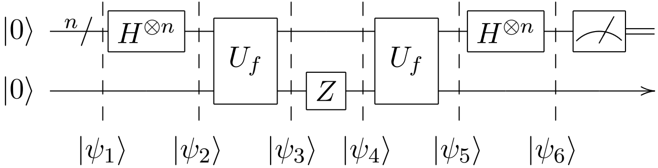
Deutsch-Josza algorithm



1 $|\psi_1\rangle = |0...0\rangle\,|0\rangle$

The Deutsch-Josza algorithm

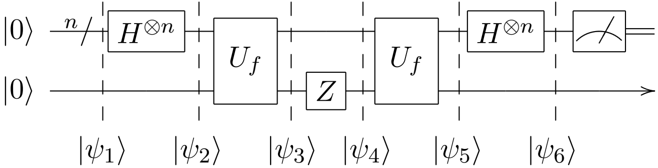
Deutsch-Josza algorithm



1. $|\psi_1\rangle = |0...0\rangle\,|0\rangle$
2. $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = \left(\sum_x |x\rangle\right)|0\rangle$

The Deutsch-Josza algorithm

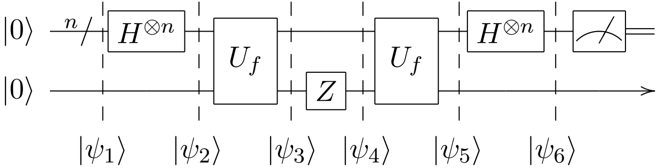
Deutsch-Josza algorithm



1. $|\psi_1\rangle = |0...0\rangle\,|0\rangle$
2. $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$
3. $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$

The Deutsch-Josza algorithm

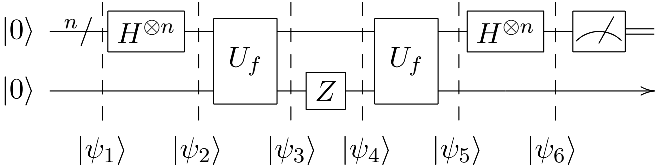
Deutsch-Josza algorithm



1. $|\psi_1\rangle = |0...0\rangle\,|0\rangle$
2. $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$
3. $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$
4. $|\psi_4\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle$

The Deutsch-Josza algorithm

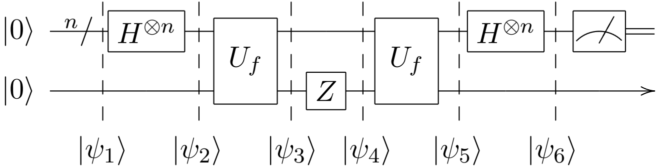
Deutsch-Josza algorithm



1. $|\psi_1\rangle = |0...0\rangle\, |0\rangle$
2. $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\, |0\rangle = (\sum_x |x\rangle)\, |0\rangle$
3. $|\psi_3\rangle = \sum_x |x\rangle\, |0 \oplus f(x)\rangle = \sum_x |x\rangle\, |f(x)\rangle$
4. $|\psi_4\rangle = \sum_x (-1)^{f(x)}\, |x\rangle\, |f(x)\rangle$
5. $|\psi_5\rangle = \sum_x (-1)^{f(x)}\, |x\rangle\, |f(x) \oplus f(x)\rangle = \sum_x (-1)^{f(x)}\, |x\rangle\, |0\rangle$

The Deutsch-Josza algorithm

Deutsch-Josza algorithm



1 $|\psi_1\rangle = |0...0\rangle\,|0\rangle$

2 $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$

3 $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$

4 $|\psi_4\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle$

5 $|\psi_5\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x) \oplus f(x)\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|0\rangle$

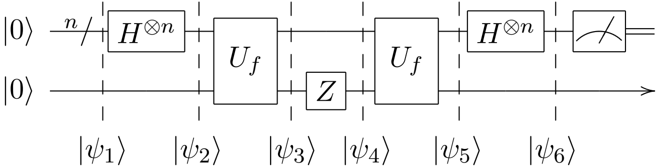
$f(x)$ constant

6 $|\psi_5\rangle = \pm \sum_x |x\rangle\,|0\rangle$

$f(x)$ balanced

The Deutsch-Josza algorithm

Deutsch-Josza algorithm



1. $|\psi_1\rangle = |0...0\rangle\,|0\rangle$
2. $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$
3. $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$
4. $|\psi_4\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle$
5. $|\psi_5\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x) \oplus f(x)\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|0\rangle$

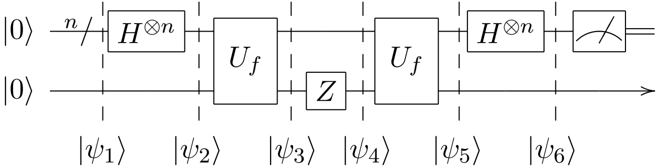
$f(x)$ constant

6. $|\psi_5\rangle = \pm \sum_x |x\rangle\,|0\rangle$
7. $|\psi_6\rangle = \pm\,|0...0\rangle\,|0\rangle$

$f(x)$ balanced

The Deutsch-Josza algorithm

Deutsch-Josza algorithm



1 $|\psi_1\rangle = |0...0\rangle\,|0\rangle$

2 $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$

3 $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$

4 $|\psi_4\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle$

5 $|\psi_5\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x) \oplus f(x)\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|0\rangle$

$f(x)$ constant

6 $|\psi_5\rangle = \pm \sum_x |x\rangle\,|0\rangle$

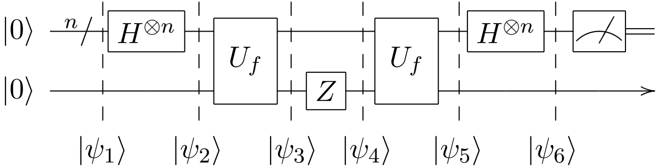
7 $|\psi_6\rangle = \pm\,|0...0\rangle\,|0\rangle$

$f(x)$ balanced

6 $|\psi_5\rangle \perp |\phi\rangle = \sum_y |y\rangle\,|0\rangle$

The Deutsch-Josza algorithm

Deutsch-Josza algorithm



1 $|\psi_1\rangle = |0...0\rangle\,|0\rangle$

2 $|\psi_2\rangle = (|0\rangle + |1\rangle)^{\otimes n}\,|0\rangle = (\sum_x |x\rangle)\,|0\rangle$

3 $|\psi_3\rangle = \sum_x |x\rangle\,|0 \oplus f(x)\rangle = \sum_x |x\rangle\,|f(x)\rangle$

4 $|\psi_4\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x)\rangle$

5 $|\psi_5\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|f(x) \oplus f(x)\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,|0\rangle$

$f(x)$ constant

6 $|\psi_5\rangle = \pm \sum_x |x\rangle\,|0\rangle$

7 $|\psi_6\rangle = \pm\,|0...0\rangle\,|0\rangle$

$f(x)$ balanced

6 $|\psi_5\rangle \perp |\phi\rangle = \sum_y |y\rangle\,|0\rangle$

7 $0 = \langle\phi|\psi_5\rangle = \langle\phi|H^{\otimes n}H^{\otimes n}|\psi_5\rangle = (\langle 0...0|\langle 0|)|\psi_6\rangle$

The Deutsch-Josza algorithm

Deutsch-Josza algorithm

- Quantum algorithm allows for deciding whether f is balanced or not with **two calls to the oracle** (independent of n)
- Query the oracle in **superposition**
- Constructive **interference** (destructive interference) yields an unity (zero) amplitude in the constant (balanced) case

The Deutsch-Josza algorithm

Deutsch-Josza algorithm

- Quantum algorithm allows for deciding whether f is balanced or not with **two calls to the oracle** (independent of n)
- Query the oracle in **superposition**
- Constructive **interference** (destructive interference) yields an unity (zero) amplitude in the constant (balanced) case

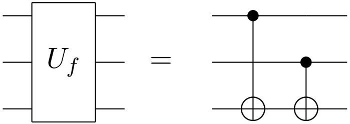
The Deutsch-Josza algorithm needs exponentially fewer calls to the oracle than the classical algorithm.

The Deutsch-Josza algorithm

Deutsch-Josza algorithm on quantum hardware

- Example for $n = 2$ input bits and the following Boolean function

x_0	x_1	$f(x_0, x_1)$
0	0	0
0	1	1
1	0	1
1	1	0

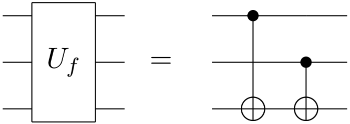


The Deutsch-Josza algorithm

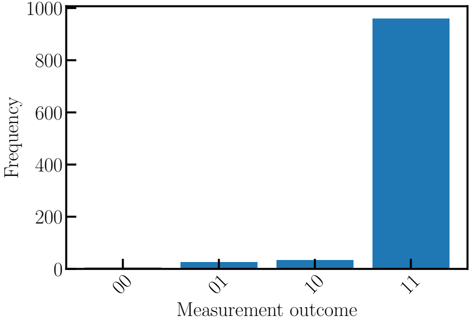
Deutsch-Josza algorithm on quantum hardware

- Example for $n = 2$ input bits and the following Boolean function

x_0	x_1	$f(x_0, x_1)$
0	0	0
0	1	1
1	0	1
1	1	0



- Results on actual quantum hardware (`ibmq_lagos`)



3.

Superdense coding

The Deutsch-Josza algorithm

Complexity theory

Hybrid quantum-classical algorithms

Complexity theory

Solving problems on a quantum computer

- Many more known quantum algorithms that (might) perform better than the best known classical algorithms
  - Shor’s factoring algorithm
  - Grover’s search algorithm
  - HHL algorithm for linear equations
  - Quantum Simulation
  - Bernstein–Vazirani algorithm
  - ...
- Exploiting quantum features such as superposition and entanglement these algorithms can outperform the best known classical algorithms

Catalog of quantum algorithms: `https://quantumalgorithmzoo.org/`

Complexity theory

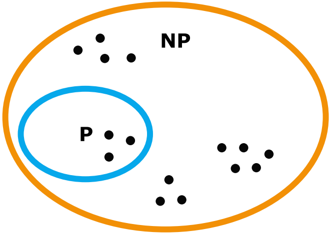
Solving problems on a quantum computer

- Many more known quantum algorithms that (might) perform better than the best known classical algorithms
  - Shor's factoring algorithm
  - Grover's search algorithm
  - HHL algorithm for linear equations
  - Quantum Simulation
  - Bernstein–Vazirani algorithm
  - ...
- Exploiting quantum features such as superposition and entanglement these algorithms can outperform the best known classical algorithms

Which problems can be solved efficiently on quantum computers?

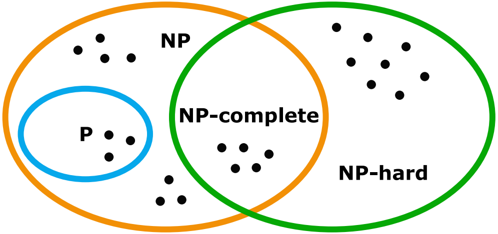
Catalog of quantum algorithms: `https://quantumalgorithmzoo.org/`

Complexity theory



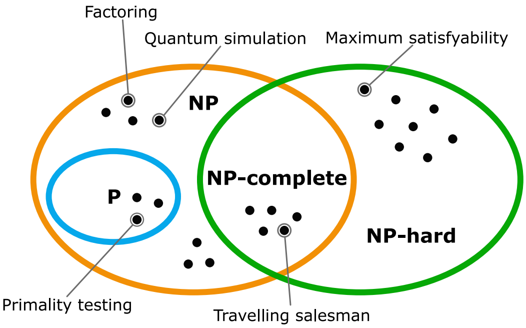
- P: decision problems solvable by a deterministic Turing machine in **polynomial time**, “easy problems”
- NP: decision problems solvable by a non-deterministic Turing machine in polynomial time
  - “Hard problems”
  - **Solution can be checked** on a deterministic Turing machine in **polynomial time**

Complexity theory



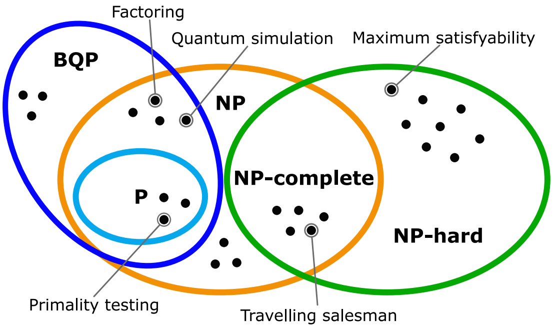
- P: decision problems solvable by a deterministic Turing machine in **polynomial time**, “easy problems”
- NP: decision problems solvable by a non-deterministic Turing machine in polynomial time
  - “Hard problems”
  - **Solution can be checked** on a deterministic Turing machine in **polynomial time**

Complexity theory



- P: decision problems solvable by a deterministic Turing machine in **polynomial time**, “easy problems”
- NP: decision problems solvable by a non-deterministic Turing machine in polynomial time
  - “Hard problems”
  - **Solution can be checked** on a deterministic Turing machine in **polynomial time**

Complexity theory



- P: decision problems solvable by a deterministic Turing machine in **polynomial time**, “easy problems”
- NP: decision problems solvable by a non-deterministic Turing machine in polynomial time
  - “Hard problems”
  - **Solution can be checked** on a deterministic Turing machine in **polynomial time**

BQP (bounded-error quantum polynomial time):

- Decision problems solvable by a quantum computer in **polynomial time**
- Quantum equivalent to P, “easy problems”

Complexity theory

The Church-Turing thesis

All **physically reasonable models of computation** have the **same set** of **computable functions**.

⇒ Quantum computers cannot compute functions that are uncomputable on a classical computer

Complexity theory

The Church-Turing thesis

All **physically reasonable models of computation** have the **same set** of **computable functions**.

⇒ Quantum computers cannot compute functions that are uncomputable on a classical computer

The extended Church-Turing thesis

All **physically reasonable models of computation differ** in complexity by **at most polynomial factors**.

⇒ Extended Church-Turing thesis would no long hold if **quantum supremacy** is demonstrated

Co

The

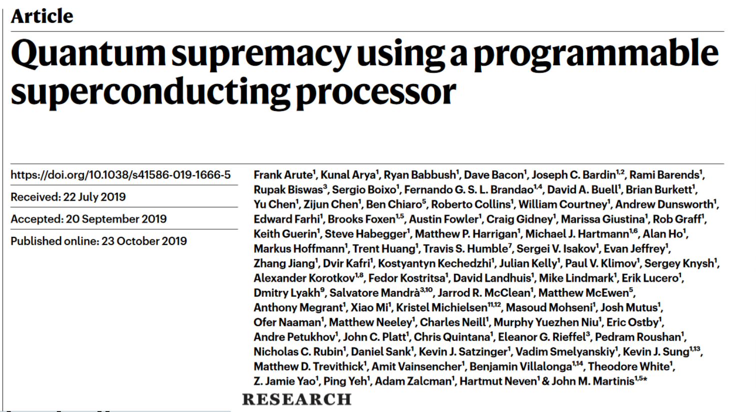
All p ... ame set of computable
func

⇒ C ... omputable on a classical
c

The

All physically reason ... t
polynomial factors.

⇒ Extended Church-T
demonstrated



Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5
Received: 22 July 2019
Accepted: 20 September 2019
Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[11,12], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas C. Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin J. Sung[1,13], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,14], Theodore White[1], Z. Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1] & John M. Martinis[1,5*]

RESEARCH

QUANTUM COMPUTING

Quantum computational advantage using photons

Han-Sen Zhong[1,2*], Hui Wang[1,2*], Yu-Hao Deng[1,2*], Ming-Cheng Chen[1,2*], Li-Chao Peng[1,2], Yi-Han Luo[1,2], Jian Qin[1,2], Dian Wu[1,2], Xing Ding[1,2], Yi Hu[1,2], Peng Hu[3], Xiao-Yan Yang[3], Wei-Jun Zhang[3], Hao Li[3], Yuxuan Li[4], Xiao Jiang[1,2], Lin Gan[4], Guangwen Yang[4], Lixing You[3], Zhen Wang[3], Li Li[1,2], Nai-Le Liu[1,2], Chao-Yang Lu[1,2†], Jian-Wei Pan[1,2†]

4.

Superdense coding

The Deutsch-Josza algorithm

Complexity theory

Hybrid quantum-classical algorithms

Hybrid quantum-classical algorithms

Current NISQ devices

- Small or intermediate scale
- Considerable amount of noise
- Only shallow circuits can be executed faithfully/no error correction
- Quantum advantage demonstrated

Hybrid quantum-classical algorithms

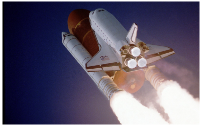
Current NISQ devices

> Small or intermediate scale
> Considerable amount of noise
> Only shallow circuits can be executed faithfully/no error correction
> Quantum advantage demonstrated

Solving "useful" problems

> Large number of qubits
> Deep circuits
> Quantum error correction necessary
> So far only proof of principle demonstrations

Hybrid quantum-classical algorithms



Current NISQ devices

- Small or intermediate scale
- Considerable amount of noise
- Only shallow circuits can be executed faithfully/no error correction
- Quantum advantage demonstrated

Solving "useful" problems

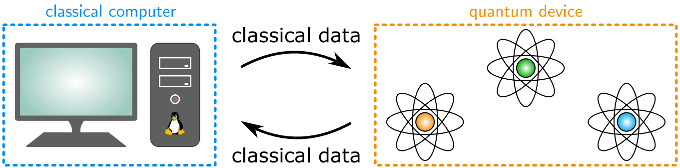
- Large number of qubits
- Deep circuits
- Quantum error correction necessary
- So far only proof of principle demonstrations

How can we utilize existing quantum hardware in a beneficial way?

Hybrid quantum-classical algorithms

Hybrid quantum-classical algorithms

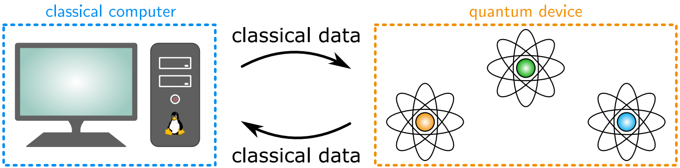
- Combine classical and quantum devices
- Rely on classical computing where possible
- Use the quantum device as a coprocessor
  - Tackle the classically hard/intractable part of the problem
  - Feed the classical data obtained from a measurement back to the classical computer



Hybrid quantum-classical algorithms

Hybrid quantum-classical algorithms

- Combine classical and quantum devices
- Rely on classical computing where possible
- Use the quantum device as a coprocessor
  - Tackle the classically hard/intractable part of the problem
  - Feed the classical data obtained from a measurement back to the classical computer



Even modest quantum hardware can yield advantages

Hybrid quantum-classical algorithms

Hybrid quantum-classical algorithms

- Focus on optimization problems

$$\min_{\vec{\theta}} \mathcal{C}(\vec{\theta}), \qquad \vec{\theta} = \mathbb{R}^n$$

- Solve them iteratively using a parametric ansatz
  - Quantum coprocessor: efficiently evaluate the cost function $\mathcal{C}(\vec{\theta}_i)$ for given $\vec{\theta}_i$
  - Classical computer: given $\mathcal{C}(\vec{\theta}_i)$, find optimized $\vec{\theta}_{i+1}$

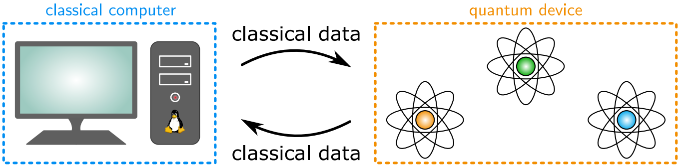
Hybrid quantum-classical algorithms

Hybrid quantum-classical algorithms

- Focus on optimization problems

$$\min_{\vec{\theta}} \mathcal{C}(\vec{\theta}), \qquad \vec{\theta} = \mathbb{R}^n$$

- Solve them iteratively using a parametric ansatz
  - Quantum coprocessor: efficiently evaluate the cost function $\mathcal{C}(\vec{\theta}_i)$ for given $\vec{\theta}_i$
  - Classical computer: given $\mathcal{C}(\vec{\theta}_i)$, find optimized $\vec{\theta}_{i+1}$



$\Rightarrow$ **Feedback loop** between the classical computer and the quantum device

Quantum Approximate Optimization Algorithm (QAOA)

Hybrid quantum-classical algorithms

Quantum Approximate Optimization Algorithm (QAOA)

> Algorithm for approximating (binary) combinatorial optimization problems

$$\min_{x \in V} C(x)$$
$$\text{subject to } x \in S$$

> x: binary string in $V = \{0,1\}^n$ encoding a solution
> $S \subseteq V$: feasible solutions
> $C : V \to \mathbb{R}$ cost function
> Objective is to find the optimal solution

Hybrid quantum-classical algorithms

The Max-Cut problem

Max-Cut

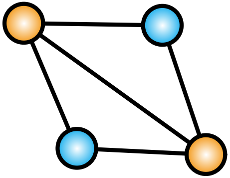
- **Input:** undirected graph $G = (V, E)$
- **Task:** find a bipartition of $V = A \cup B$ such that the number of edges between A and B is maximal

Hybrid quantum-classical algorithms

The Max-Cut problem

Max-Cut

- **Input:** undirected graph $G = (V, E)$
- **Task:** find a bipartition of $V = A \cup B$ such that the number of edges between A and B is maximal



- Max-Cut is NP-hard

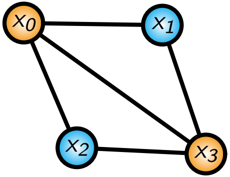
$\Rightarrow$ We cannot find a (quantum) algorithm which solves it polynomial time

- We can however try to find a good approximation to the exact solution in polynomial time

Hybrid quantum-classical algorithms

Max-Cut as combinatorial optimization problem

- Max-Cut on a Graph $G = (V, E)$ can be expressed as combinatorial optimization problem
- Label the vertices as x_i define a function w_{ij}

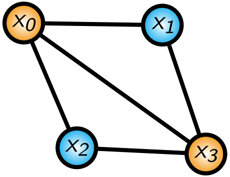


$$x_i = \begin{cases} 0 & \text{if } i \in A \\ 1 & \text{if } i \in B \end{cases} \qquad w_{ij} = \begin{cases} 1 & \text{iff } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Hybrid quantum-classical algorithms

Max-Cut as combinatorial optimization problem

- Max-Cut on a Graph $G = (V, E)$ can be expressed as combinatorial optimization problem
- Label the vertices as x_i define a function w_{ij}



$$x_i = \begin{cases} 0 & \text{if } i \in A \\ 1 & \text{if } i \in B \end{cases} \qquad w_{ij} = \begin{cases} 1 & \text{iff } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- Cost function

$$C(x) = \sum_{i,j=0}^{n-1} w_{ij} x_i (x_j - 1) = \sum_{(i,j)\in E} \left(x_i(x_j - 1) + x_j(x_i - 1) \right)$$

⇒ Contribution of -1 iff endpoints of edge (i, j) belong to different subsets

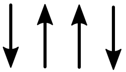
- Finding the Max-Cut for G is equivalent to minimizing $C(x)$

Hybrid quantum-classical algorithms

Max-Cut as Hamiltonian problem

- Cost function can be turned into a Hamiltonian using the mapping $x_i \to \frac{1}{2}(1 - Z_i)$

$$H_c = \frac{1}{2} \sum_{(i,j)\in E} (Z_i Z_j - 1)$$



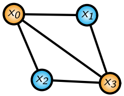
- Diagonal Hamiltonian of Ising type, summands commute

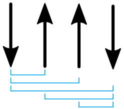
Hybrid quantum-classical algorithms

Max-Cut as Hamiltonian problem

- Cost function can be turned into a Hamiltonian using the mapping $x_i \to \frac{1}{2}(1 - Z_i)$

$$H_c = \frac{1}{2} \sum_{(i,j) \in E} (Z_i Z_j - 1)$$





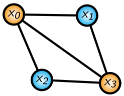
- Diagonal Hamiltonian of Ising type, summands commute

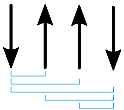
Hybrid quantum-classical algorithms

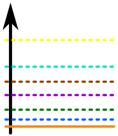
Max-Cut as Hamiltonian problem

- Cost function can be turned into a Hamiltonian using the mapping $x_i \to \frac{1}{2}(1 - Z_i)$

$$H_c = \frac{1}{2} \sum_{(i,j)\in E} (Z_i Z_j - 1)$$







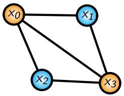
- Diagonal Hamiltonian of Ising type, summands commute
- The eigenstates of H are computational basis states encode graph cuts
- The lower the energy, the larger the number of edges between the subsets
- The ground state encodes the bit string of the optimal solution x^*

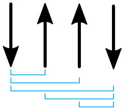
Hybrid quantum-classical algorithms

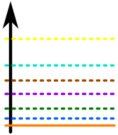
Max-Cut as Hamiltonian problem

- Cost function can be turned into a Hamiltonian using the mapping $x_i \to \frac{1}{2}(1 - Z_i)$

$$H_c = \frac{1}{2} \sum_{(i,j)\in E} (Z_i Z_j - 1)$$







- Diagonal Hamiltonian of Ising type, summands commute
- The eigenstates of H are computational basis states encode graph cuts
- The lower the energy, the larger the number of edges between the subsets
- The ground state encodes the bit string of the optimal solution x^*

How to obtain a low energy state of H?

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- We want to find a parametric quantum state $|\psi_p(\vec{\gamma},\vec{\beta})\rangle$, $\vec{\gamma}$, $\vec{\beta} \in \mathbb{R}^p$ which minimizes

$$\mathcal{C}(\vec{\gamma},\vec{\beta}) = \left\langle \psi_p(\vec{\gamma},\vec{\beta}) \middle| H_c \middle| \psi_p(\vec{\gamma},\vec{\beta}) \right\rangle$$

E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- We want to find a parametric quantum state $|\psi_p(\vec{\gamma},\vec{\beta})\rangle$, $\vec{\gamma}$, $\vec{\beta} \in \mathbb{R}^p$ which minimizes

$$\mathcal{C}(\vec{\gamma},\vec{\beta}) = \left\langle \psi_p(\vec{\gamma},\vec{\beta}) \middle| H_c \middle| \psi_p(\vec{\gamma},\vec{\beta}) \right\rangle$$

- Transverse field Hamiltonian

$$H_x = \sum_i X_i$$

- Ansatz for $|\psi_p(\vec{\gamma},\vec{\beta})\rangle$

$$|\psi_p(\vec{\gamma},\vec{\beta})\rangle = e^{-i\beta_p H_x} e^{-i\gamma_p H_c} \dots e^{-i\beta_1 H_x} e^{-i\gamma_1 H_c} |+\rangle^{\otimes n}$$

- $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is an eigenstate of X, $X|+\rangle = +1|+\rangle$

E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

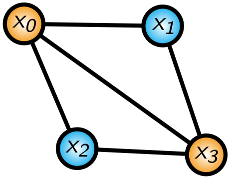
Hybrid quantum-classical algorithms

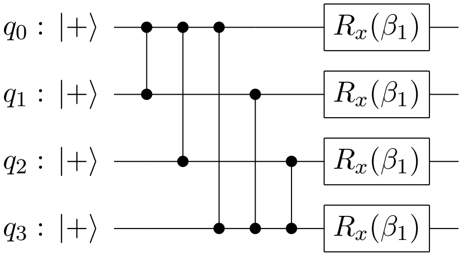
The Quantum Approximate Optimization Algorithm (QAOA)

- Ansatz for $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$

$$|\psi_p(\vec{\gamma}, \vec{\beta})\rangle = e^{-i\beta_p H_x} e^{-i\gamma_p H_c} \dots e^{-i\beta_1 H_x} e^{-i\gamma_1 H_c} |+\rangle^{\otimes n}$$

- Circuit for $p = 1$



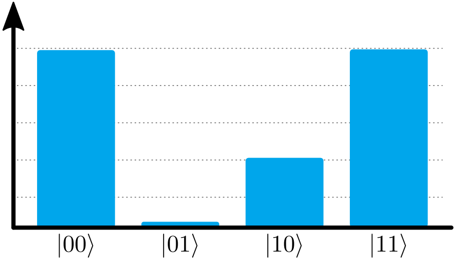


E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ can be an (entangled) superposition of basis states
- Measuring $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ gives us distribution of bit strings x



E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- After minimizing $\mathcal{C}(\vec{\gamma}, \vec{\beta})$ the wave function $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ has dominant component(s) of low energy states of H_c

$\Rightarrow$ Measuring $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ gives us a bit string(s) x with high approximation ratio

$$\alpha = \frac{C(x)}{C(x^*)}$$

E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

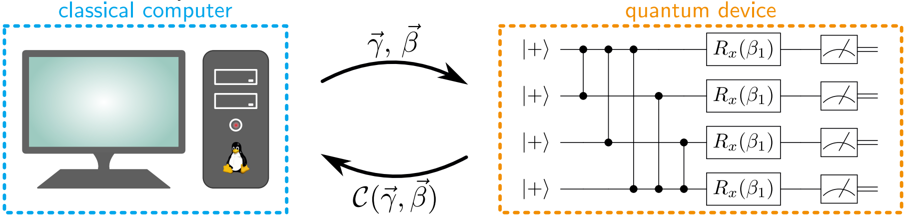
The Quantum Approximate Optimization Algorithm (QAOA)

- After minimizing $\mathcal{C}(\vec{\gamma}, \vec{\beta})$ the wave function $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ has dominant component(s) of low energy states of H_c

$\Rightarrow$ Measuring $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ gives us a bit string(s) x with high approximation ratio

$$\alpha = \frac{C(x)}{C(x^*)}$$

- Optimization of the parameters



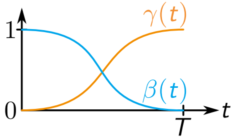
E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- Ansatz is inspired by trotterized adiabatic time evolution
- Choose functions $\gamma(t)$, $\beta(t)$ such that

$$\gamma(t) \to \begin{cases} 0 & \text{for } t \to 0 \\ 1 & \text{for } t \to T \end{cases} \qquad \beta(t) \to \begin{cases} 1 & \text{for } t \to 0 \\ 0 & \text{for } t \to T \end{cases}$$



and set

$$\gamma_p = \gamma(p\Delta t)\Delta t, \qquad \beta_p = \beta(p\Delta t)\Delta t$$

- With these conventions each time steps corresponds to

$$e^{-i\beta_p H_x} e^{-i\gamma_p H_c} \approx e^{-i\left(\beta(p\Delta t)H_x + \gamma(p\Delta t)H_c\right)\Delta t}$$

- In the limit $p \to \infty$ the QAOA ansatz can be seen as adiabatic transformation of the initial eigenstate of H_x to the ground state of H_c

E. Farhi, J. Goldstone, S. Gutmann, arXiv:1411.4028

Hybrid quantum-classical algorithms

The Quantum Approximate Optimization Algorithm (QAOA)

- Even $p = 1$ can in general not be simulated on a classical computer efficiently
- For some problems classical algorithms got a better approximation ratio α
- Performance depends on the ratio of variables to clauses

$\Rightarrow$ Theoretically it is not entirely clear how QAOA performs

B. Barak et al., arXiv:2106.05900
E. Farhi, A. Harrow, arXiv:1602.07674
V. Akshay, H Philathong, M. E. S. Morales, J. D. Biamonte, Phys. Rev. Lett 124, 090504 (2020)

Variational Quantum Eigensolver (VQE)

Hybrid quantum-classical algorithms

Variational Quantum Algorithms

- Similar principle can be used to find ground states of quantum Hamiltonians
- Define a cost function

$$\mathcal{C}(\vec{\theta}) = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle$$

- $|\psi(\vec{\theta})\rangle$ ansatz realized by a parametric quantum circuit
- Provided $|\psi(\vec{\theta})\rangle$ is expressive enough the minimum of $\mathcal{C}(\vec{\theta})$ is obtained for the ground state of H

Peruzzo et al., Nat. Commun. 5, 1 (2014)
J. R. McClean et al., New J. Phys. 18, 023023 (2016)

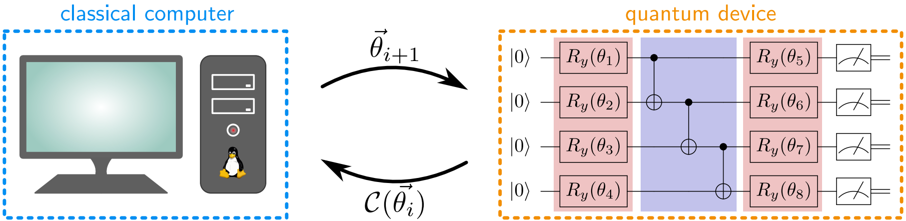
Hybrid quantum-classical algorithms

Variational Quantum Algorithms

- Similar principle can be used to find ground states of quantum Hamiltonians
- Define a cost function

$$\mathcal{C}(\vec{\theta}) = \langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle$$

- $|\psi(\vec{\theta})\rangle$ ansatz realized by a parametric quantum circuit
- Provided $|\psi(\vec{\theta})\rangle$ is expressive enough the minimum of $\mathcal{C}(\vec{\theta})$ is obtained for the ground state of H



Peruzzo et al., Nat. Commun. 5, 1 (2014)
J. R. McClean et al., New J. Phys. 18, 023023 (2016)

Variational Quantum Algorithms

Variational Quantum Eigensolver (VQE)

- For Hamiltonians H that are the sum of a polynomial number of terms the cost function can be measured efficiently on a quantum device, for example

$$H = \sum_{i=1}^{N} h_{i,i+1} \qquad \Leftrightarrow \qquad \mathcal{C}(\theta) = \sum_{i} \langle \psi(\vec{\theta}) | h_{i,i+1} | \psi(\vec{\theta}) \rangle$$

- In general $\langle \psi(\vec{\theta}) | h_{i,i+1} | \psi(\vec{\theta}) \rangle$ cannot be efficiently evaluated on a classical computer

A. Kandala et al., Nature 549, 242 (2017)
J. R. McClean et al., New J. Phys. 18, 023023 (2016)
Peruzzo et al., Nat. Commun. 5, 1 (2014)

Variational Quantum Algorithms

Variational Quantum Eigensolver (VQE)

- For Hamiltonians H that are the sum of a polynomial number of terms the cost function can be measured efficiently on a quantum device, for example

$$H = \sum_{i=1}^{N} h_{i,i+1} \qquad \Leftrightarrow \qquad \mathcal{C}(\theta) = \sum_{i} \langle \psi(\vec{\theta}) | h_{i,i+1} | \psi(\vec{\theta}) \rangle$$

- In general $\langle \psi(\vec{\theta}) | h_{i,i+1} | \psi(\vec{\theta}) \rangle$ cannot be efficiently evaluated on a classical computer
- Example: the Ising model

$$H_{\text{Ising}} = \sum_{i=1}^{N-1} Z_i Z_{i+1} + h \sum_{i=1}^{N} X_i$$

A. Kandala et al., Nature 549, 242 (2017)
J. R. McClean et al., New J. Phys. 18, 023023 (2016)
Peruzzo et al., Nat. Commun. 5, 1 (2014)

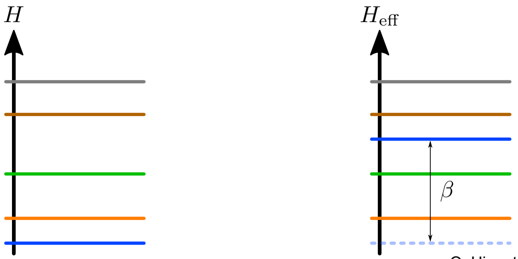
Variational Quantum Algorithms

Variational Quantum Deflation (VQD)

- VQE can also be extended to higher excited states

$$H_{\text{eff}} = H + \beta \sum_{k=0}^{n} |\psi_k\rangle\langle\psi_k|$$

- β has to be chosen large enough



O. Higgot, D. Wang, S. Brieley, Quantum 3, 156 (2019)

Variational Quantum Algorithms

Variational Quantum Deflation (VQD)

- In practice we never explicitly construct the projector
- Rather we have have to computer overlaps

$$\mathcal{C}(\vec{\theta}) = \langle \psi(\vec{\theta}) | H_{\text{eff}} | \psi(\vec{\theta}) \rangle = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle + \beta \sum_k |\langle \psi_k | \psi(\vec{\theta}) \rangle|^2$$

- There are several ways to compute the overlap
  - “Reversing the circuit”: double the depth but keep same number of qubits
  - SWAP test: same circuit depth but double the number of qubits

O. Higgot, D. Wang, S. Brieley, Quantum 3, 156 (2019)

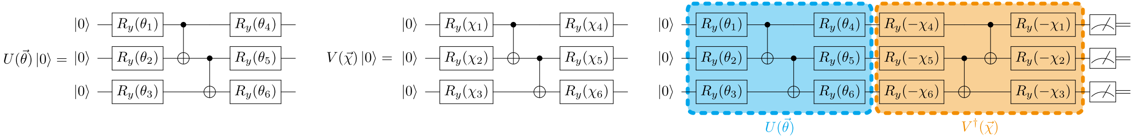
Variational Quantum Algorithms

Reversing the circuit for computing absolute values of overlaps

- Since both $|\psi_k\rangle = V(\vec{\chi})\,|0\rangle$ and $\psi(\vec{\theta}) = U(\vec{\theta})\,|0\rangle$ are variational circuits the absolute value of the overlap is given by

$$|\langle\psi_k|\psi(\vec{\theta})\rangle| = |\,\langle 0|\,V^\dagger(\vec{\chi})U(\vec{\theta})\,|0\rangle\,|$$

- The absolute value can thus be estimated by preparing the state $V^\dagger(\vec{\chi})U(\vec{\theta})\,|0\rangle$ and recording the number of times one measures $|0\rangle$ at the end
- For commonly used quantum gates the inverse is often trivial

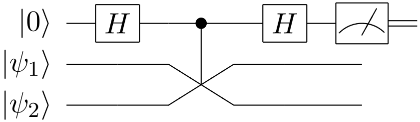


O. Higgot, D. Wang, S. Brieley, Quantum 3, 156 (2019)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers

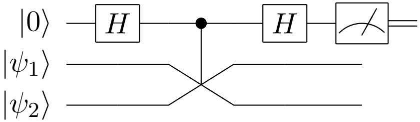


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate

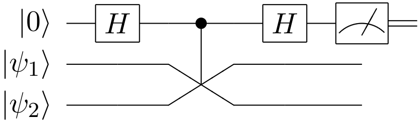


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate
  1. Start from $|0, \psi, \phi\rangle$

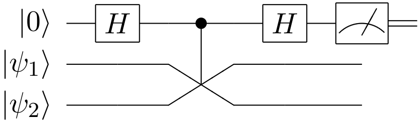


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate
  1. Start from $|0,\psi,\phi\rangle$
  2. Hadamard creates $\frac{1}{2}(|0,\psi,\phi\rangle + |1,\psi,\phi\rangle)$

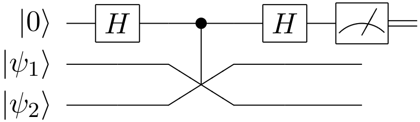


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate
  1. Start from $|0, \psi, \phi\rangle$
  2. Hadamard creates $\frac{1}{2}(|0, \psi, \phi\rangle + |1, \psi, \phi\rangle)$
  3. Fredkin gate yields $\frac{1}{2}(|0, \psi, \phi\rangle + |1, \phi, \psi\rangle)$

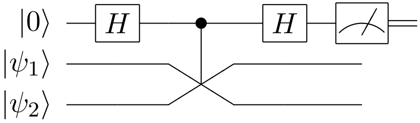


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate
  1. Start from $|0,\psi,\phi\rangle$
  2. Hadamard creates $\frac{1}{2}(|0,\psi,\phi\rangle + |1,\psi,\phi\rangle)$
  3. Fredkin gate yields $\frac{1}{2}(|0,\psi,\phi\rangle + |1,\phi,\psi\rangle)$
  4. Hadamard creates $\frac{1}{2}|0\rangle(|\psi,\phi\rangle + |\psi,\phi\rangle) + \frac{1}{2}|1\rangle(|\psi,\phi\rangle - |\phi,\psi\rangle)$



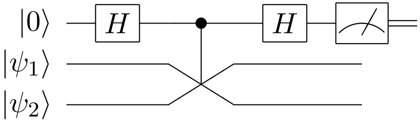
A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate

1. Start from $|0,\psi,\phi\rangle$
2. Hadamard creates $\frac{1}{2}(|0,\psi,\phi\rangle + |1,\psi,\phi\rangle)$
3. Fredkin gate yields $\frac{1}{2}(|0,\psi,\phi\rangle + |1,\phi,\psi\rangle)$
4. Hadamard creates $\frac{1}{2}|0\rangle(|\psi,\phi\rangle + |\psi,\phi\rangle) + \frac{1}{2}|1\rangle(|\psi,\phi\rangle - |\phi,\psi\rangle)$
5. Probability of measuring the ancilla in state $|0\rangle$: $p_0 = \frac{1}{2} + \frac{1}{2}|\langle\psi,\phi\rangle|^2$

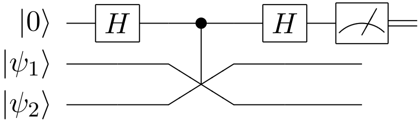


A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)

Variational Quantum Algorithms

SWAP test for computing absolute values of overlaps

- The SWAP test allows for computing the overlap of arbitrary states in registers
- Based on the controlled SWAP operation/Fredkin gate
  1. Start from $|0, \psi, \phi\rangle$
  2. Hadamard creates $\frac{1}{2}(|0, \psi, \phi\rangle + |1, \psi, \phi\rangle)$
  3. Fredkin gate yields $\frac{1}{2}(|0, \psi, \phi\rangle + |1, \phi, \psi\rangle)$
  4. Hadamard creates $\frac{1}{2}\,|0\rangle\,(|\psi, \phi\rangle + |\psi, \phi\rangle) + \frac{1}{2}\,|1\rangle\,(|\psi, \phi\rangle - |\phi, \psi\rangle)$
  5. Probability of measuring the ancilla in state $|0\rangle$: $p_0 = \frac{1}{2} + \frac{1}{2}|\,|\psi, \phi\rangle\,|^2$



- More efficient ways for VQD: destructive SWAP test

A. Barenco *et al.*, SIAM Journal on Computing 26, 1541 (1997)
L. Cincio *et al.*, New J. Phys. 20, 113022 (2018)
J. C. Garcia-Escartin, P. Chamorro-Posada Phys. Rev. A 87, 052330 (2013)

Variational Quantum Algorithms

QAOA

- Combinatorial optimization problems
- Problem Hamiltonian is diagonal in the computational basis
- Circuit structure is fixed
- In the limit of infinite layers provably converges to the exact solution

Variational Quantum Algorithms

QAOA

- Combinatorial optimization problems
- Problem Hamiltonian is diagonal in the computational basis
- Circuit structure is fixed
- In the limit of infinite layers provably converges to the exact solution

VQE

- Ground states/low-lying excitations
- Efficient as long as H has only a polynomial number of terms
- Hamiltonian exists only as a measurement
- Great freedom choosing the circuit
  - Problem requirements
  - Available hardware
  - Expressiveness

Variational Quantum Algorithms

QAOA

- Combinatorial optimization problems
- Problem Hamiltonian is diagonal in the computational basis
- Circuit structure is fixed
- In the limit of infinite layers provably converges to the exact solution

VQE

- Ground states/low-lying excitations
- Efficient as long as H has only a polynomial number of terms
- Hamiltonian exists only as a measurement
- Great freedom choosing the circuit
  - Problem requirements
  - Available hardware
  - Expressiveness

- Best answer for the given set of resources
- Largely resilient to systematic errors of the device

Example: VQE for the Schwinger model

Variational Quantum Algorithms

VQE for the Schwinger model

- Hamiltonian in spin language after integrating out the gauge fields

$$H = w \sum_{j1}^{N-1} \left(\sigma_j^+ \sigma_{j+1}^- + \sigma_j^- \sigma_{j+1}^+ \right) + \frac{m}{2} \sum_{j=1}^{N} (-1)^j Z_j + g \sum_{j=1}^{N} \left(\varepsilon_0 - \frac{1}{2} \sum_{l=1}^{j} \left(Z_j + (-1)^j \right) \right)^2$$

- Symmetries of the problem
  - Conservation of total charge $Q_{\text{tot}} = \sum_n Q_n = \sum_n Z_n$
  - CP symmetry within a charge sector: reflecting around the center and flipping the spins
- Symmetries can be incorporated in the VQE ansatz

C. Kokail et al., Nature 569, 355 (2019)

Variational Quantum Algorithms

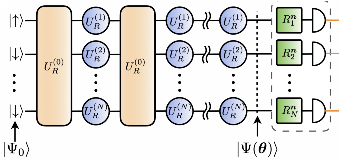
VQE for the Schwinger model

- Resource Hamiltonians used for gate operations $U = \exp(-i\theta H_R^k)$

$$H_R^0 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}(\sigma_i^+ \sigma_j^- + \sigma_i^- \sigma_j^+) + B \sum_{i=1}^{N} Z_i, \qquad H_R^j = \frac{\Delta_0}{2} Z_j$$

- Resource Hamiltonians respect charge conservation
- Restrict the parameters of the single-qubit gates as $\theta^j = -\theta^{N+1-j}$ to enforce CP symmetry
- For large systems the bulk should approximately be translation invariant

$\Rightarrow$ Restrict parameters in bulk to the same value



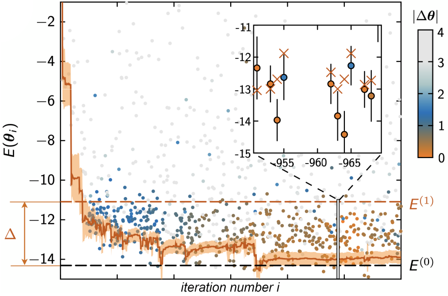
- Initial wave function compatible with the symmetries: Neel state $|\psi_0\rangle = |\uparrow\downarrow \cdots \uparrow\downarrow\rangle$

C. Kokail et al., Nature 569, 355 (2019)

Variational Quantum Algorithms

VQE for the Schwinger model

- Results on a trapped ion quantum computer with 20 qubits formed by $^{40}Ca^{+}$ ions
- Simulation involves 6 layers and 15 parameters



⇒ Ground state can be determined with good accuracy

C. Kokail et al., Nature 569, 355 (2019)

Variational Quantum Algorithms

VQE for the Schwinger model

- As we have seen in the previous lecture, a negative mass in the Schwinger model corresponds to the case of $\theta = \pi$
- For this value of θ, the model undergoes a second order phase transition for
- The phase structure with negative masses can also be explored with VQE

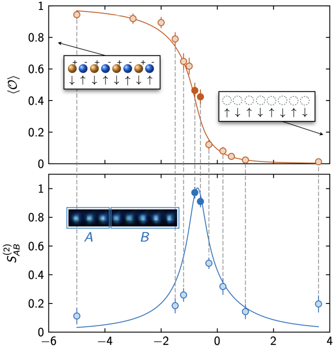
C. Kokail et al., Nature 569, 355 (2019)

Variational Quantum Algorithms

VQE for the Schwinger model

- As we have seen in the previous lecture, a negative mass in the Schwinger model corresponds to the case of $\theta = \pi$
- For this value of θ, the model undergoes a second order phase transition for
- The phase structure with negative masses can also be explored with VQE
- The detect the phase transition monitor

$$O = \frac{1}{2N(N-1)} \sum_{i,j>i} \left((1 + (-1)^i Z_i)(1 + (-1)^j Z_j) \right)$$



C. Kokail et al., Nature 569, 355 (2019)

Thank you for your attention!

Questions?